University of Zurich[UZH]

Department of Informatics

A Smart Thermostat for Demand-Side Management With Real-Time Electricity Prices

Dissertation submitted to the
Faculty of Business, Economics and Informatics
of the University of Zurich

to obtain the degree of
Doktor der Wissenschaften, Dr. sc.
(corresponds to Doctor of Science, PhD)

presented by
Michael Shann
from Zurich, ZH, Switzerland

approved in October, 2019

at the request of
Prof. Dr. Sven Seuken
Prof. Alex Rogers, Ph.D.

The Faculty of Business, Economics and Informatics of the University of Zurich hereby authorizes the printing of this dissertation, without indicating an opinion of the views expressed in the work.

Zurich, October 23, 2019

Chairman of the Doctoral Board: Prof. Dr. Thomas Fritz

Abstract

One of the biggest challenges of this century is transforming the way we produce and consume energy to make our society sustainable and limit the impact of climate change. This transformation will necessitate the de-carbonization of many aspects of our lives because human-made emissions of greenhouse gases like CO2 are believed to be the main cause of climate change. One solution that has been proposed is to move away from fossil fuels to cleaner energy sources like wind and solar, and to electrify energy-intense sectors of our economy like heating and transportation. However, large amounts of renewables in the power grid might threaten the stability of the grid. To solve this problem, *demand-side management* (DSM) has been proposed as a mechanism to make the demand for energy more flexible and let it adapt to the available supply.

In this thesis, I consider the scenario of *home heating with real-time electricity prices*. This case study fits well into de-carbonization plans since home heating is a major factor of energy consumption, home heating could efficiently be done by electric heat pumps, and real-time pricing is one of the approaches for DSM. The thesis investigates the feasibility and effectiveness of DSM with real-time pricing for home heating. For this, I propose a *smart thermostat* that learns a user's heating preferences and automatically heats the house. The smart thermostat uses a machine learning algorithm to learn how a user wants to trade off comfort and cost, and computes a sequentially optimal heating policy that takes the uncertainty from future weather and electricity market conditions into account. The smart thermostat was evaluated in a *field experiment* involving 30 users over a period of 30 days. The results show that overall, the smart thermostat enabled users to successfully manage their heating given real-time prices. Moreover, machine learning simplified the interaction with a real-time electricity market, compared to a thermostat without machine learning. Regarding the effectiveness of DSM with real-time pricing, the experimental data shows that the users' settings led to a large amount of demand-response, reducing the peak-hour energy consumption by 38% compared to off-peak hours.

Zusammenfassung

Eine der größten Herausforderungen dieses Jahrhunderts besteht darin, die Art und Weise, wie wir Energie produzieren und verbrauchen, zu verändern, um unsere Gesellschaft nachhaltig zu gestalten und die Auswirkungen des Klimawandels zu begrenzen. Diese Transformation wird die Entkarbonisierung vieler Aspekte unseres Lebens erforderlich machen, denn die vom Menschen verursachten Emissionen von Treibhausgasen wie CO2 gelten als Hauptursache für den Klimawandel. Eine Lösung, die vorgeschlagen wurde, besteht darin, von fossilen Brennstoffen zu saubereren Energiequellen wie Wind und Sonne überzugehen und gleichzeitig energieintensive Sektoren unserer Wirtschaft wie Gebäudeheizung und Verkehr zu elektrifizieren. Allerdings können große Mengen an erneuerbaren Energien im Stromnetz die Stabilität des Netzes gefährden. Um dieses Problem zu lösen, wurde *Energie-Laststeuerung* (englisch: demand-side management, DSM) vorgeschlagen, um die Energienachfrage flexibler zu gestalten und sie an das verfügbare Angebot anzupassen.

In dieser Arbeit betrachte ich das Szenario der *Hausheizung mit Echtzeit-Strompreisen.* Diese Fallstudie passt gut zu den Entkarbonisierungsplänen, da das Heizen von Häusern ein wichtiger Faktor für den Energieverbrauch ist, die Hausheizung effizient mit elektrischen Wärmepumpen betrieben werden kann und die Echtzeit-Preisgestaltung einer der Ansätze für DSM ist. Die Arbeit untersucht die Machbarkeit und Wirksamkeit von DSM mit Echtzeit-Preisen für die Hausheizung. Dazu schlage ich einen *intelligenten Thermostaten* vor, der die Heizpräferenzen eines Benutzers lernt und das Haus automatisch heizt. Der intelligente Thermostat verwendet einen maschinellen Lernalgorithmus, um zu lernen, wie ein Benutzer Komfort und Kosten gegen einander abwägt, und berechnet eine sequentiell optimale Heizstrategie, welche die Unsicherheit durch zukünftige Wetter- und Strommarktbedingungen berücksichtigt.

Der intelligente Thermostat wurde in einem *Feldexperiment* mit 30 Benutzern über einen Zeitraum von 30 Tagen ausgewertet. Die Ergebnisse zeigen, dass der intelligente Thermostat es den Nutzern insgesamt ermöglichte, ihre Heizung bei Echtzeitpreisen erfolgreich zu steuern. Darüber hinaus vereinfachte das maschinelle Lernen die Interaktion mit einem Echtzeit-Strommarkt im Vergleich zu einem Thermostat ohne maschinelles Lernen. Was die Effektivität von DSM mit Echtzeit-Preisen betrifft, so zeigen die experimentellen Daten, dass die Einstellungen der Benutzer zu einer großen Nachfragereduktion (englisch: demand response) führten, wodurch der Energieverbrauch zu Hauptlastzeiten um 38% gegenüber Normalzeiten reduziert wurde.

Acknowledgments

I would like to thank my advisor Sven Seuken for providing me the opportunity to conduct research on a topic that I find extraordinarily important. Sven was always supportive of my ideas and tirelessly helped improve them. He has taught me to be a rigorous scientist. The result of processing an idea with Sven's scientific filter is usually an exceptionally clear and meaningful thought that has a high signal-to-noise ratio. This filter is invaluable, and I will take a part of it with me. I am also impressed by the breadth and depth of his knowledge, which makes it very stimulating to work with him.

I thank Alex Rogers for serving as an external reviewer of this thesis and providing very valuable feedback on my research. Alex's work on putting the "smarts" into the smart grid has been an important source of inspiration for my own research.

I thank Alper Alan, Enrico Costanza, and Sarvapali Ramchurn for the collaboration on "smarthermo" and for offering me to visit the Agents, Interaction, and Complexity research group at the University of Southampton. Planning and conducting the field experiment in Southampton was definitely one of the highlights of my PhD.

I thank the members of the Computation and Economics research group at the University of Zurich for creating an enjoyable and motivating environment, with many deep but also funny discussions which I believe only happen when the crazy and creative minds of highly intelligent scientists are involved. Thank you Gianluca Brero, Ludwig Dierks, Timo Mennle, Dmitry Moor, and Steffen Schuldenzucker.

I thank all the other nice people of the Department of Informatics, but especially Cosmin Basca, Lorenz Fischer, Sofia Hotomsky, Martina Huber, Sarah Mennicken, Bibek Paudel, Philipp Stutz, and Mihaela Verman for all the fun hours we spent together.

I gratefully acknowledge the financial support of the Swiss National Science Foundation.

Lastly, I thank my wife Luba for all her support, patience, and love throughout these years. Her sharp graphic design eye and her InDesign skills helped me quite a few times during paper and poster writing sessions. For sure, this thesis is to a great extent her merit.

Contents

1 Introduction and Overview of Results

1.1 Motivation[1]

The power of science and technology has led to an incredible economic growth over the last 250 years that has helped people in many countries escape famine, illness, early death and other scourges of humanity. This exponential growth has increased the quality of life for billions of people, and it is not unreasonable to believe that science and technology will increase the quality of life even further in the future (Harari, 2015). As a consequence of this unprecedented growth, the demand for energy has increased exponentially as well. Most of this energy has been delivered in the form of fossil fuels like coal, oil, and gas. However, burning fossil fuels on such a large scale has led to serious problems because it releases enormous amounts of climate-active gases like CO2 into the atmosphere where they aggravate the greenhouse effect to such levels that global warming and climate change are becoming a reality that will negatively affect many of us. Nowadays, there is a large body of scientific evidence that supports the hypothesis that mankind is mainly responsible for global warming (Goudie, 2018). Therefore, it is clear that there is an urgent need for action.

Against this background, most governments around the world have agreed to start efforts to reduce greenhouse gas emissions and to transition to low-carbon economies. For example, the main goal of the Paris agreement from 2015, which most countries have signed to date, is to limit "global temperature increase to well below 2 degrees Celsius, while pursuing efforts to limit the increase to 1.5 degrees [in the long term]."[2] In the context of the Paris agreement, many countries state ambitious goals to reduce the emission of greenhouse gases. For example, the European Union and Switzerland have committed themselves to reduce their emissions by 40% and 50% by 2030 compared to the levels of 1990, respectively. An important implication of the Paris agreement is that if the global temperature has to be stabilized, the net greenhouse gas emissions have to

[1] Please note that this chapter borrows freely from my own previous work in (Shann and Seuken, 2013), (Shann and Seuken, 2014), and (Shann et al., 2017).

[2] Press release of United Nations Framework Convention on Climate Change(UNFCCC): https://unfccc.int/process-and-meetings/the-paris-agreement/what-is-the-paris-agreement

converge to 0 in the long run, from which a remaining greenhouse gas emission "budget" can be calculated (Christensen, 2018).
To achieve these aims, the way we produce and consume energy has to change radically. It seems inevitable that our current main energy source - fossil fuels - will have to be replaced by renewable and less CO2-intense energy sources like solar, wind, or hydropower. This would mean that in the future, energy-intense sectors of our economy like transportation and house heating would have to be increasingly powered by renewable energy sources. One way to achieve this is to base our mobility on electric vehicles and to heat our homes and offices by electric heat pumps. Electrifying our economy with renewable energy sources is a plan that, together with a globally increased demand for energy,[3] presents a number of challenges to the electricity grid. Most importantly, stabilizing the grid becomes more and more of a challenge. To maintain the stability of the grid, demand and supply have to be balanced within tight bounds. If this cannot be guaranteed, brownouts (voltage drops) or even blackouts (power outages) are possible. Currently, the structure of electricity markets is such that supply follows demand. That means if there is an increase in demand, then some power plant has to produce more energy to satisfy this demand. Conversely, if demand decreases, then the energy producers have to lower their production within a short period of time. However, the production level of renewable energy sources is hard to control due to their volatile nature. Therefore, as the share of renewable energy sources keeps growing, matching supply and demand will become increasingly difficult, which might seriously threaten the stability of the grid (Cramton and Ockenfels, 2012).

1.1.1 Demand-side Management

Part of a solution to address the problem of grid stability could be to manage the demand side by incentivizing consumers to adapt their consumption levels to the amount of energy available in the grid (Ramchurn et al., 2012). *Demand-side management* (DSM) includes all technological and economic measures that need to be taken to ensure that the consumption of electricity is always balanced with its production, such as remote controlling devices or providing economic incentives for customers to decrease their consumption in times where energy supply is scarce, and make the most efficient use of energy in times where it is abundant (Palensky and Dietrich, 2011). Table 1.1 presents a summary of possible DSM methods.
One way to encourage consumers to decrease their demand when energy is scarce is via

[3] An increase of 40% is projected by the International Energy Agency by 2040 compared to 2018; see `https://www.iea.org/weo2018/`

DSM method	Explanation	Incentive
Direct Control	Utility can remote control customer's equipment	rate discount
Curtailable Loads	Customer has to reduce load	rate discount
Demand Bidding	Customer can bid for curtailing	rate discount
Capacity Market	Customer has to reduce load during scarcity events	rate discount
Time-of-use Pricing	Fixed price schedule	variable price
Critical Peak Pricing	Fixed, high rates during scarcity events	variable price
Real-time Pricing	Customer faces whole-sale prices	variable price

Table 1.1: Methods for demand-side management. For detailed explanations see (Palensky and Dietrich, 2011; Albadi and El-Saadany, 2007).

financial incentives (Albadi and El-Saadany, 2007). A particular financial mechanism that has been put forward is *real-time pricing*. With real-time pricing, electricity is priced in regular, short time intervals according to demand and supply. Real-time pricing has a number of advantages over flat pricing. First, economists argue that real-time pricing improves system reliability and mitigates market power in the long term (Barbose et al., 2004). Second, it offers consumers the opportunity to save significant amounts of money if they are willing to dynamically adjust their consumption (Chiles et al., 2015; Faruqui and Palmer, 2011). A number of power companies in the US and Europe have successfully conducted pilot studies to assess the potential benefits and the feasibility of using real-time pricing for *residential users* (see e.g., (Hammerstrom et al., 2007; Kärkkäinen et al., 2004; King, 2010; Summit Blue Consulting, 2006)). Some power companies already offer real-time pricing programs to their users.[4]

1.1.2 Home Heating and Smart Thermostats

While energy plays a large role in many domains, residential heating is a big driver of energy consumption, accounting for approximately 45% and 62% of the total household energy consumption in the US and the UK, respectively, which amounts to 10% and 18% of the respective country's total energy consumption (Palmer and Cooper, 2013; U.S. Energy Information Administration, 2013). With the goal in mind to move away from fossil fuels, the electrification of heating using heat pumps is seen as a key technology for achieving a society that is more sustainable. Indeed, many low-carbon scenarios assume that in the future, a majority of houses will be heated by heat pumps (see, e.g., (Committee on Climate Change, 2013)). These reasons make home heating a formidable

[4] E.g., Commenwealth Edison's "Residential Real-time Pricing Program": `https://rrtp.comed.com/`

case study to explore the potential for DSM with real-time electricity prices.

Designing a home heating system in which users participate in a real-time energy market presents several challenges. Obviously, it is not feasible for users to constantly monitor the energy price and manually adjust their thermostat whenever prices change. Thus, there is a need for an intelligent agent, which I call the *smart thermostat*, that automatically reacts to price changes on the user's behalf. To be able to act autonomously, an intelligent agent needs to have an internal model of the environment it acts in and a metric that it can use to select an action depending on the input it receives (Wooldridge, 2009). The design proposal for this thesis is to use *utility-based agents* that model the user's preferences with a utility function (Russell and Norvig, 2003), an approach grounded in micro-economic theory (Mas-Colell et al., 1995). A utility-based agent will take actions in order to maximize the user's utility. One advantage of the utility-based approach is that it allows us to quantify trade-offs between conflicting goals (Russell and Norvig, 2003).

Reasoning about trade-offs is important in the context of DSM because the user naturally needs to weigh the value derived from consuming electricity against the costs of doing so. In the heating domain, the fundamental trade-off is between comfort (heating to a particular temperature) and cost (for heating to that temperature) at different price levels. Some users might be willing to spend a lot of money to have their home always heated to a comfortable temperature, while others may want to decrease their temperature if energy becomes too expensive. This means that to achieve high economic efficiency, it must be possible to personalize the smart thermostat to individual users.

Because of this user heterogeneity, it is essential for the smart thermostat to *learn the user's preferences* to account for the individual differences that exist between different people. Learning the preferences needs to be both intuitive and unobtrusive for the user. The key challenge lies in the fact that for the user it is often difficult to reason explicitly about his utility function. In the context of home heating, manually specifying how to trade off comfort and cost at all price levels might lead to high cognitive costs, which might not be desirable because it could decrease the economic efficiency of the interaction between user and the market. Therefore, the user should be allowed to express his needs in a natural way that keeps the interaction complexity at the necessary minimum.

Equipped with the knowledge about its owner's preferences, the main task of the smart thermostat is to heat the house. To be able to act rationally, i.e., maximize the user's utility, it has to compute a sequential heating policy that can be used to determine the course of action. Since the smart thermostat acts in a stochastic environment where future states are not fully known in advance, the policy needs to take into account any uncertainty that comes from future weather conditions and electricity prices. In this context, employing utility functions is particularly useful because it allows the computation of the *expected utility* of an action, the sum of the utilities over all possible outcomes weighted by the probability of occurrence (Russell and Norvig, 2003).

An important consideration for a smart thermostat is how to design the user interface (UI). To make good choices when setting the temperature, the user needs to be able to visually explore the impact that potential decisions could have on his comfort and his heating bill. However, on a more fundamental level, the question is how to expose the user to the smartness of the system. The learning part of the thermostat makes automatic inferences about the user's preferences based on previous inputs. Therefore, in a certain way, it potentially *aggregates all inputs* the user has provided to the system. This might lead to situations where the user's currently desired set point and the algorithmically determined optimal temperature differ. It is the task of a good UI to defuse such situations by mediating between the user and the algorithmic components of the smart thermostat.

Summarizing, a smart thermostat for DSM with real-time electricity prices consists of three main components:

1. a model of and an algorithm to learn the user's preferences for heating his home,
2. an algorithm to compute a sequential heating policy that maximizes the user's utility and takes the uncertainty from future environmental conditions into account, and
3. a suitable UI that facilitates the interaction between the user and the smart thermostat.

1.2 Goal and Research Questions of the Thesis

The goal of this thesis is to design a smart thermostat for DSM with real-time electricity prices. I follow the approach outlined in the previous section and divide this task into three subtasks that deal with the three components that are necessary to implement a smart thermostat. Along these lines, the research questions that this thesis answers deal with the three components and the validation of the design:

Question 1. How to model the user's preferences for home heating, and how to learn these preferences efficiently and unobtrusively?

Question 2. How to compute a heating policy that maximizes the user's utility and takes the uncertainty from future environmental conditions into account?

Question 3. How to design a suitable UI that facilitates the interaction between the user and the smart thermostat?

Question 4. How do people react to and use the smart thermostat? Does the learning algorithm improve the usability of the smart thermostat, compared to a non-learning thermostat?

1.3 Related Work

The task of designing a smart thermostat is a complex and inter-disciplinary endeavor. Accordingly, this thesis draws on prior research that has been conducted in a diverse set of areas, which can be categorized into four main topics: 1. Automated Control in the Smart Grid, 2. Home Heating, 3. Preference Elicitation, and 4. Hidden Market Design. In the following, I will give an overview of each of these four areas of research.

Automated Control in the Smart Grid. Rogers et al. (2012a) provide an introduction to the smart grid from a multi-agent systems perspective, while Ramchurn et al. (2012) describe the opportunities for AI research in this field. Vytelingum et al. (2010) study micro-storage management for the smart grid, and devise agent strategies that automatically react to price changes. However, they assume that the amount of energy each user desires per time period is known in advance, and thus the problem of eliciting users' preferences does not arise in their model. Jia and Tong (2012) consider the retailer's perspective, and provide a solution for optimal pricing of energy, given that users trade off comfort for cost. However, they also do not consider how a DSM system could learn the user's preferences. While there has been much research on how to design algorithms for DSM in the smart grid, experimental research that studies how people would react to such systems is relatively sparse. Notable examples are the following studies. Yang and Newman (2013) examine the real-world uptake of a smart thermostat with 23 participants. They highlight how sub-optimal decisions taken by a smart thermostat are likely to cause user frustrations and may lead them to abandon the technology. Bourgeois et al. (2014) deploy energy-aware washing machines in 18 households and find that sending suggestions on when to do the laundry via text messages is more effective than other interventions. Costanza et al. (2014) conduct a field experiment with 10 participants that used "Agent B", an agent that helps users book their washing machine given real-time prices. Their results indicate that users are willing to shift their washing in response to real-time prices. Alan et al. (2014) test "Tariff Agent", an agent that helps users select electricity tariffs on a daily

basis, in a field experiment with 10 users. The results show that people are willing to delegate decisions regarding energy consumption to an agent.
My smart thermostat differs from the above systems in two key ways. First, it is *fully autonomous*, i.e., it takes decisions on the user's behalf instead of just giving advice to the user. Second, the system's decisions have a direct impact on users' well-being via the temperature it sets in their respective homes, while previous systems only affected the study participants' financial rewards.

Home Heating. Developing algorithms for energy efficient heating is an active area of research both in academia and the industry. One direction of research tries to optimize the heating with the help of predictions of future environmental conditions that affect the heating such as the outside temperature. The goal is to compute a heating plan, typically for the next 24-72 hours, *that minimizes the energy costs*, subject to the constraint that the thermal comfort lies within some acceptable boundaries. To do so, one needs a thermal model of the house, a method to predict future environmental conditions, and the actual optimization routine.
Various approaches have been studied for all three components. For example, Rogers et al. (2011) use Gaussian processes to predict the external temperature, while Oldewurtel et al. (2010) use Kalman filters. Proposed methods for the optimization include integer programming (Rogers et al., 2011), reinforcement learning (Yu and Dexter, 2010), fuzzy logic control (Homod et al., 2012), and model predictive control (MPC) (McLaughlin et al., 2012; Oldewurtel et al., 2010). MPCs are online algorithms that iteratively solve an optimization problem for a given time horizon to find the best sequence of (continuous) control actions, but only apply the first action to the system. After each time step, the system state is observed and a new optimization problem is solved given the new state. Thus, the time horizon is shifted one time step into the future. Cigler et al. (2012) estimate that MPCs can lead to a reduction in the energy consumption of 15-30%.
Another research direction is to develop algorithms that try to sense and predict the occupancy of the house with the goal of reducing the inside temperature when people are not at home. For example, Scott et al. (2011) use motion sensing and machine learning to find patterns in user behavior to heat adaptively. Occupancy detection has also been applied in commercial thermostats. For example, the Nest thermostat has a motion sensor that detects people's presence.[5] It tries to learn a heating schedule that conforms to its users' habits in order to save energy by decreasing the temperature when people are not at home ("auto away" feature). Tado is another commercial thermostat that exploits the

[5] https://nest.com

user's smartphone GPS sensors to predict occupancy patterns in order to save energy.[6] While there has been considerable research on home heating, especially on the optimization aspect, the research community has paid less attention to the problem of learning the user's preferences with regard to heating. In particular, the trade-off between comfort and cost that arise with dynamic energy prices have been largely ignored.

Preference Elicitation. My work on preference elicitation draws on previous work by Chajewska et al. (2000) and Boutilier (2002) who defines preference elicitation as the "process of extracting the necessary preference or utility information from a user". "Necessary information" in this context means the information an intelligent agent needs in order to act on the user's behalf. Good preference elicitation methods need to have a model of the user's preferences (i.e., a formal model of the utility function), a set of questions to ask the user in order to update the knowledge of the utility function, a query criterion to select the next question, and a termination criterion to decide whether enough information about the user has been gathered to end the elicitation process.
Chajewska et al. (2000) use Bayesian inference to update the knowledge of the utility function. They define the next query to be the one that maximizes the expected value of information, and they terminate the elicitation process if the expected regret (i.e., the difference in expected utility when taking their recommended action vs. the optimal action) is below a certain threshold. The smart thermostat also uses Bayesian inference to update the parameters of the utility function. However, while they consider a domain where arbitrary queries can be synthesized, I consider the problem of selecting the best query from a stream of potential queries, which is called selective sampling or stream-based sampling (Settles, 2009).

Hidden Market Design. When a user needs to interact with a real-time energy market to heat his house, there is a tension between economic efficiency and cognitive costs, because for the user, it can be overwhelming to think about the optimal set point for different price levels, which might lead to sub-optimal decisions on his side. To address this tension, I use the *hidden market design* paradigm introduced by Seuken et al. (2010b), who argued that it is often necessary to hide some of the market's complexity from the users. They showed that a hidden market UI can reduce the interaction complexity for the users, while still maintaining the loop between the market and the users (Seuken et al., 2010c). One way to achieve this goal is to design a *learning agent* that operates in the background and mediates between the user and the market. The goal of implementing this agent is to

[6] `https://www.tado.com`

reduce the cognitive costs for the user, while still keeping the important feedback loop between the user and the market that is needed for economic efficiency. Seuken et al. (2010a) present a case study on how to apply hidden market design to the design of a peer-to-peer backup market, demonstrating that it is possible to hide a significant amount of complexity from the user, while still keeping the important user-market loop.

1.4 Publications Contained in this Thesis

This thesis consists of three papers, each of which addresses one of the research questions. I will recap the research questions stated earlier and list the corresponding paper that answers the research question:

Question 1. How to model the user's preferences for home heating, and how to learn these preferences efficiently and unobtrusively?
Publication: Shann, M. and Seuken, S. (2013). An Active Learning Approach to Home Heating in the Smart Grid. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, IJCAI '13, Beijing, China.

Question 2. How to compute a heating policy that maximizes the user's utility?
Publication: Shann, M. and Seuken, S. (2014). Adaptive Home Heating under Weather and Price Uncertainty using GPs and MDPs. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '14, Paris, France.

Question 3. How to design a suitable UI that facilitates the interaction between the user and the smart thermostat?
Question 4. How do people react to and use the smart thermostat? Does the learning algorithm improve the usability of the smart thermostat, compared to a non-learning thermostat?
Publication: Shann, M., Alan, A., Seuken, S., Costanza, E., and Ramchurn, S. D. (2017). Save Money or Feel Cozy? A Field Experiment Evaluation of a Smart Thermostat that Learns Heating Preferences. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '17, São Paulo, Brazil.

While research questions 1 and 2 are theoretical problems that were addressed by designing algorithms that were validated with simulations, research question 3 and especially research question 4 are different in that they are empirical. Therefore, to answer questions

3 and 4, it was necessary to build, deploy and test a real-world system in a *field experiment*. Note that from the results obtained by the field experiment, a fourth paper was published. However, this paper is not included in this thesis since I am not the first author of the paper. The paper was published at the following venue:

Publication: Shann, M., Alan, A., Seuken, S., Costanza, E., and Ramchurn, S. D. (2016). It is too Hot: An In-Situ Study of Three Designs for Heating. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, San Jose, USA.

1.5 Summary of Contributions

In the following, I will provide a short summary of my three papers.

1.5.1 An Active Learning Approach to Home Heating in the Smart Grid

Inherent to the home heating problem is the need for the user to trade off *comfort* with the *costs* of heating. I model this trade-off using a utility function that is composed of a *value function* and a *cost function*. The value function quantifies the user's level of comfort for an indoor temperature (i.e., his willingness to pay for this temperature), while the cost function quantifies how expensive it is to heat the house to a certain temperature at the current price of energy. The user's utility for a certain indoor temperature at a certain price is then the difference between his value function and the cost function.

The value function has two parameters that can differ from person to person. Every user is assumed to have a single *most-preferred temperature*, to which he would heat his house if energy were for free. Additionally, the user is assumed to experience some degree of discomfort if the actual indoor temperature deviates from the most preferred temperature. His discomfort increases the further the temperature deviates from his most preferred temperature. The strength of this effect, which I call the user's *sensitivity* to temperature deviations, can vary from user to user. The most-preferred temperature and the sensitivity are the two parameters of the utility function that the smart thermostat has to learn over time to be able to heat optimally on the user's behalf.

To learn the parameters, I propose an active learning algorithm that can ask the user at most once a day: "What is your preferred temperature *now*, at price p?" The one-query-per-day restriction is motivated by the goal of designing a non-intrusive interaction. If

the user decides to answer this question by providing a temperature value, the algorithm can use this information to update its knowledge of the parameters of the utility function. This is done via *Bayesian inference*. The algorithm assumes that the user implicitly maximizes his utility when answering the question. The temperature which the user enters into the smart thermostat is treated as a *noisy sample* because the algorithm expects the user to make slight mistakes when reasoning about his optimal temperature.
The remaining question for the algorithm is how to select the best query from the stream of prices it encounters. I propose an algorithm that asks in such a way as to minimize the expected cumulative loss in utility the user will suffer. This cumulative loss is approximated by the cumulative loss the user will suffer until the end of the next day (i.e., until the end of the current day plus one additional day). To find the optimal query time, the algorithm computes an *optimal stopping policy* $\pi(t, p_t) \rightarrow \{sample, continue\}$. For each time t and price p_t, this policy prescribes whether to ask the user for feedback now, or whether to wait. This policy can be computed using dynamic programming (Peskir and Shiryaev, 2006).
I compare my algorithm against two state-of-the-art learning algorithms from the literature: one that queries the user to maximize the information gain (MacKay, 1992), and another one that queries to user to minimize the expected predictive loss with respect to the optimal temperature T^{opt} (Cohn et al., 1996). The performance metric used is the user's cumulative loss, i.e., the difference between the utility the user would have had if the algorithm had known the user's true preferences and the utility he actually had. The simulation results show that my algorithm significantly outperform the baselines from the literature.

1.5.2 Adaptive Home Heating under Weather and Price Uncertainty using GPs and MDPs

In the second paper, I formalize the home heating problem in order to be able to compute a sequentially optimal heating policy that maximizes the user's utility, taking into account the uncertainty coming from future electricity prices and weather conditions. I model the home heating problem as a Markov decision process (MDP). Three components are important to formulate the MDP:

1. a model of the thermal dynamics of a house,
2. a prediction of future environmental conditions (i.e., weather and electricity prices),
3. and a model of the user's preferences.

Points 1. and 2. are necessary because the algorithm has to know how its actions (i.e., heat or not) will affect the future state of its control variable, which in our case is the indoor temperature. To model the thermal dynamics of the house I use an approach that has been successfully applied in prior research (Rogers et al., 2012b). To make predictions of future environmental conditions, I use Gaussian processes (GPs), which have been used before in this context (Rogers et al., 2011). My technical contribution is a method to derive the state transition function of the MDP by using the predictive distributions of the GPs. The home heating MDP can then be instantiated using these predictive distributions.

Regarding point 3., the MDP uses the utility function presented in the previous section as a reward function, which ensures that the algorithm will optimally heat the house for the user, given that the user's preferences are known. This paper is not concerned with learning the preferences, but instead assumes that these are known sufficiently well. Solving the MDP yields a sequentially optimal heating policy that can be used to heat the house in a way that maximizes the user's utility.

To evaluate the proposed heating MDP, I conduct two simulation experiments. In Experiment 1, I compare the MDP to a conventional thermostat and a mixed-integer program that tries to maximize the user's utility but does not consider the uncertainty in the predictions of the external temperature and the prices (Rogers et al., 2011). The performance metric is the user's cumulative utility. The MDP significantly outperforms these two baseline algorithms because it is the only one that takes into account the uncertainty in future temperatures and prices explicitly and consequently can better adapt to changing conditions.

In Experiment 2, I compare the MDP to model predictive control (MPC), a state-of-the-art method used in home heating (Oldewurtel et al., 2010; McLaughlin et al., 2012). The simulation results show an interesting trade-off between computational run-time and performance. If both algorithms have little time (i.e. a few seconds) to compute a solution, the MPC performs better (achieves higher user utility on average), while the MDP can perform better if the time that is given to the algorithms to find a solution increases to a few minutes. Summarizing, the simulation results show that the MDP beats some state-of-the-art algorithms from the literature, while there is an interesting trade-off between run-time and performance when comparing the MDP and the MPC approach.

1.5.3 A Field Experiment Evaluation of a Smart Thermostat that Learns Heating Preferences

After laying the algorithmic foundation for designing a smart thermostat that learns the user's preferences and optimally heats based on these preferences, I wanted to know how people would actually interact with such an intelligent agent. There are several challenges when implementing such a system. Participating in a market with real-time energy prices is a fairly complex task. Even though the smart thermostat heats autonomously, the user needs to understand the concept of real-time prices and what this means for him. He has to understand how a thermostat works and how real-time prices affect the temperature in his home. Because if not, the user might set up the thermostat in a way that might leave him dissatisfied with the system.

However, the main challenge lies in the fact that there is an inherent tension between the user input (i.e., a set point temperature) and the machine learning output. The learning algorithm assumes the user input (i.e., the set point changes) to be noisy data. Thus, when the user changes the temperature, the algorithm will update the parameters of the utility function. However, the optimal temperature according to the model might be a different value than what the user just provided.

To address this issue, I designed and tested two UIs based on two different interaction paradigms. The first one is based on direct manipulation and exposes the user directly to the workings of the machine learning algorithm, while the second one hides the inner workings of the learning algorithm. In the first UI, called *learning direct UI*, the set point that is displayed is always the learned set point by the thermostat. In contrast, in the second UI, the *learning indirect UI*, if the user changes the set point, it temporarily overrides the learned set point for one hour and then switches back to the learned set point.

These two UIs were tested in a field experiment with 30 participants in the UK over the course of 30 days. Each participant got a budget of £100 for heating in a virtual energy market with real-time prices derived from the UK electricity spot market. The experimental setup was that every day, the heating costs in the virtual market would be subtracted from the participants' virtual heating budget, and at the end of the study, the participants could keep whatever budget they had left as an experimental reward. The participants were divided into three test groups with 10 users each: one for each of the two learning UIs, and one group that used a manual version of the thermostat (without machine learning algorithm).

The results of the experiment show that the majority of users were satisfied with the smart thermostat, and trusted it to automatically adjust the temperature for them. More importantly, the data shows that the machine learning algorithm increased the usability of the system, compared to manual non-learning version of the thermostat. This is true for the learning indirect UI. Moreover, a detailed quantitative analysis of the economic behavior of the 30 participants shows that the users reacted to price changes in an economically rational way, and on average, they were willing to decrease their indoor temperature by 3 °C when energy was most expensive. However, due to the thermal inertia of the homes, the indoor temperature did not decrease by more than 1 °C, even during peak price hours. Still, this price-sensitive behavior led to a large amount of demand response, reducing the average energy consumption by 38 % during peak hours.

1.6 Conclusion and Future Work

One of the biggest challenges of this century is transforming the way we produce and consume energy to make our societies sustainable and limit the impact of climate change. This will necessitate the de-carbonization of many aspects of our lives, including the heating of our houses. It is still unclear how we will interact with energy in the future, but it seems likely that if a substantial part of the energy mix comes from renewable and volatile sources, making the supply of energy more transparent to users using methods of DSM might become increasingly important. In this case, smart devices that use principles of AI and assist users in a (semi-)autonomous fashion could play a key role in future electricity markets.

The goal of this thesis was to explore the potential of DSM with real-time pricing in the domain of home heating. I have proposed and tested a smart thermostat that learns the user's heating preferences and heats automatically to maximize the user's utility. The results of the field experiment show that it is indeed possible to deploy such a system in real homes, get user acceptance, and induce large amounts of demand response. However, the results of the field experiment also indicate that handling real-time prices for heating presents a significant challenge to many users. Not only is the subject of heating itself relatively complex, but combining it with real-time pricing adds a level of complexity that could potentially overwhelm many users, despite best efforts to design simple and intuitive UIs. This suggests that although real-time pricing has the potential to create economically efficient market interactions, it might not be the best DSM method for everyone, but only for users who are willing and able to deal with its complexities.

1.6.1 Limitations

The work presented in this thesis has a number of limitations. First, the user model might not be as expressive as required for a real-world application. The model uses the indoor temperature as a proxy for the overall thermal comfort of a person. However, while the temperature is arguably one of the more important factors determining the thermal comfort, it is known that many additional factors such as relative humidity, airspeed, and a person's metabolic rate influence thermal comfort (Fanger, 1970). Moreover, the utility function assumes that a person's utility depends linearly on the price of energy, which seems a reasonable first approximation. However, it is well possible that preferences are piece-wise constant in the price of energy: the user is willing to pay a bit more to keep the current comfort level until the cost becomes too expensive at a certain threshold. Another aspect my user model ignores is time: user preferences might depend on the time of the day and the day of the week.

Secondly, the thermal model of the house is relatively simple. It does not capture the thermal properties of the house and the physical process of heating in detail. For the field experiment, this model was enough to create the necessary sense of realism such that the participants could immerse themselves into the scenario of heating with real-time prices. However, for a real-world application, more accurate thermal models might be required to make useful cost predictions.

1.6.2 Future Work

Future research could design and test a smart thermostat that includes a heating optimization algorithm. This would create a few interesting HCI challenges since for the user it might be even more difficult to understand the decisions taken by the thermostat. Moreover, future deployments of a smart thermostat could incorporate more sophisticated user models that capture user preferences more accurately, for example as elaborated by Auffenberg et al. (2017).

However, the most important goal for future research will be to produce high-quality evidence on the long-term feasibility and effectiveness of different methods for DSM. This will help companies and policymakers base their decisions on a solid foundation. The design space for possible methods is enormous, and every method has its distinct advantages and disadvantages. The work presented in this thesis suggests that AI-powered smart thermostats might make DSM effective and, at the same time, feasible for users who are willing to participate in a real-time energy market.

Bibliography

Alan, A., Costanza, E., Fischer, J., Ramchurn, S. D., Rodden, T., and Jennings, N. R. (2014). A Field Study of Human-agent Interaction for Electricity Tariff Switching. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems*, Paris, France.

Alan, A. T., Shann, M., Costanza, E., Ramchurn, S. D., and Seuken, S. (2016). It is Too Hot: An In-Situ Study of Three Designs for Heating. In *Proceedings of the 2016 SIGCHI Conference on Human Factors in Computing Systems*, Santa Clara, California, USA.

Albadi, M. and El-Saadany, E. (2007). Demand Response in Electricity Markets: An Overview. In *Power Engineering Society General Meeting, 2007. IEEE*, pages 1--5.

Auffenberg, F., Snow, S., Stein, S., and Rogers, A. (2017). A Comfort-Based Approach to Smart Heating and Air Conditioning. *ACM Trans. Intell. Syst. Technol.*, 9(3):28:1--28:20.

Barbose, G., Goldman, C., and Neenan, B. (2004). *A Survey of Utility Experience with Real Time Pricing*. Lawrence Berkeley National Laboratory.

Bourgeois, J., van der Linden, J., Kortuem, G., Price, B. A., and Rimmer, C. (2014). Conversations with My Washing Machine: An In-the-wild Study of Demand Shifting with Self-generated Energy. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '14, Seattle, Washington.

Boutilier, C. (2002). A POMDP Formulation of Preference Elicitation Problems. In *Eighteenth National Conference on Artificial Intelligence*, Edmonton, Alberta, Canada. American Association for Artificial Intelligence.

Chajewska, U., Koller, D., and Parr, R. (2000). Making Rational Decisions Using Adaptive Utility Elicitation. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, Austin Texas.

Chiles, T., Shutika, K., and Coleman, P. (2015). Riding the Electricity Market as an Energy Management Strategy: Savings from Real-Time Pricing. *Lawrence Berkeley National Laboratory. LBNL Report: LBNL-1004321.*

Christensen, J. (2018). *Emissions Gap Report 2018*. UNEP DTU Partnership.

Cigler, J., Prívara, S., Váňa, Z., Žáčeková, E., and Ferkl, L. (2012). Optimization of Predicted Mean Vote Index Within Model Predictive Control Framework: Computationally Tractable Solution. *Energy and Buildings*, 52:39--49.

Cohn, D. A., Ghahramani, Z., and Jordan, M. I. (1996). Active Learning with Statistical Models. *Journal of Artificial Intelligence Research*, 4:129--145.

Committee on Climate Change (2013). Fourth Carbon Budget Review. Technical Report.

Costanza, E., Fischer, J. E., Colley, J. A., Rodden, T., Ramchurn, S. D., and Jennings, N. R. (2014). Doing the Laundry with Agents: A Field Trial of a Future Smart Energy System in the Home. In *Proceedings of the 2014 SIGCHI Conference on Human Factors in Computing Systems*, Toronto, Ontario, Canada.

Cramton, P. and Ockenfels, A. (2012). Economics and Design of Capacity Markets for the Power Sector. *Zeitschrift fuer Energiewirtschaft*, 36(2):113--134.

Fanger, P. (1970). *Thermal comfort: analysis and applications in environmental engineering*. McGraw-Hill.

Faruqui, A. and Palmer, J. (2011). Dynamic Pricing of Electricity and its Discontents. *Regulation*, 34(2).

Goudie, A. S. (2018). *Human Impact on the Natural Environment*. Wiley.

Hammerstrom, D., Ambrosio, R., Brous, J., Carlon, T., Chassin, D., DeSteese, J., Guttromson, R., Horst, G., Järvegren, O., Kajfasz, R., et al. (2007). Pacific Northwest Gridwise Testbed Demonstration Projects. *Part I. Olympic Peninsula Project.*

Harari, Y. N. (2015). *Sapiens: A Brief History of Humankind*. Harper.

Homod, R. Z., Sahari, K. S. M., Almurib, H. A., and Nagi, F. H. (2012). Gradient Autotuned Takagi--Sugeno Fuzzy Forward Control of a HVAC System Using Predicted Mean Vote Index. *Energy and Buildings*, 49:254--267.

Jia, L. and Tong, L. (2012). Optimal Pricing for Residential Demand Response: A Stochastic Optimization Approach. In *Proceedings of the Allerton Conference on Communication, Control and Computing*.

Kärkkäinen, S., Ikäheimo, J., Kofod, C., Giraud, D., Nordvik, H., and Grande, O. (2004). Energy Efficiency and Load Curve Impacts of Commercial Development in Competitive Markets-Results from the EFFLOCOM Pilots: VTT, Finnland.

King, C. (2010). PowerCentsDC Program Final Report. Smart Meter Pilot Program, Inc., eMeter Corp. Washington, DC.

MacKay, D. J. C. (1992). Information-Based Objective Functions for Active Data Selection. *Neural Computation*, 4(4):590--604.

Mas-Colell, A., Whinston, M. D., Green, J. R., et al. (1995). *Microeconomic Theory*. Oxford University Press.

McLaughlin, Z. Y. L., Jia, L., Murphy-Hoye, M. C., Pratt, A., and Tong, L. (2012). Modeling and Stochastic Control for Home Energy Management. In *Proceedings of the Power and Energy Society (PES) General Meeting*.

Oldewurtel, F., Parisio, A., Jones, C. N., Morari, M., Gyalistras, D., Gwerder, M., Stauch, V., Lehmann, B., and Wirth, K. (2010). Energy Efficient Building Climate Control Using Stochastic Model Predictive Control and Weather Predictions. In *Proceedings of the American Control Conference*.

Palensky, P. and Dietrich, D. (2011). Demand Side Management: Demand Response, Intelligent Energy Systems, and Smart Loads. *Industrial Informatics, IEEE Transactions on*, 7(3):381--388.

Palmer, J. and Cooper, I. (2013). Great Britain's Housing Energy Fact File. Departmant of Energy and Climate Change (DECC).

Peskir, G. and Shiryaev, A. (2006). *Optimal Stopping and Free-Boundary Problems*. Lectures in Mathematics ETH Zürich. Birkhäuser Verlag.

Ramchurn, S. D., Vytelingum, P., Rogers, A., and Jennings, N. R. (2012). Putting the 'Smarts' into the Smart Grid: A Grand Challenge for Artificial Intelligence. *Commun. ACM*, 55(4):86--97.

Rogers, A., Maleki, S., Ghosh, S., and Nicholas R, J. (2011). Adaptive Home Heating Control Through Gaussian Process Prediction and Mathematical Programming. In *Proceedings of the International Workshop on Agent Technology for Energy Systems (ATES)*.

Rogers, A., Ramchurn, S. D., and Jennings, N. R. (2012a). Delivering the Smart Grid: Challenges for Autonomous Agents and Multi-agent Systems Research. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, Toronto, Ontario, Canada.

Rogers, A., Wilcock, R., Ghosh, S., and Jennings, N. R. (2012b). A Scalable Low-cost Solution to Provide Personalized Home Heating Advice to Households. In *Proceedings of the Fourth ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, BuildSys '12, Toronto, Ontario, Canada.

Russell, S. J. and Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. Pearson Education.

Scott, J., Bernheim Brush, A., Krumm, J., Meyers, B., Hazas, M., Hodges, S., and Villar, N. (2011). PreHeat: Controlling Home Heating Using Occupancy Prediction. In *Proceedings of the 13th International Conference on Ubiquitous Computing*, UbiComp '11, Beijing, China.

Settles, B. (2009). Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin--Madison.

Seuken, S., Charles, D., Chickering, M., and Puri, S. (2010a). Market Design Analysis for a P2P Backup System. In *Proceedings of the 11th ACM Conference on Electronic Commerce*, Cambridge, Massachusetts, USA.

Seuken, S., Jain, K., and Parkes, D. C. (2010b). Hidden Market Design. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI'10, Atlanta, Georgia.

Seuken, S., Jain, K., Tan, D. S., and Czerwinski, M. (2010c). Hidden Markets: UI Design for a P2P Backup Application. In *Proceedings of the 2010 SIGCHI Conference on Human Factors in Computing Systems*, Atlanta, Georgia, USA.

Shann, M., Alan, A., Seuken, S., Costanza, E., and Ramchurn, S. D. (2017). Save Money or Feel Cozy? A Field Experiment Evaluation of a Smart Thermostat That Learns

Heating Preferences. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '17, Sao Paulo, Brazil.

Shann, M. and Seuken, S. (2013). An Active Learning Approach to Home Heating in the Smart Grid. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, IJCAI '13, Beijing, China.

Shann, M. and Seuken, S. (2014). Adaptive Home Heating Under Weather and Price Uncertainty Using GPS and MDPs. In *Proceedings of the 13th International Conference on Autonomous Agents and Multi-agent Systems*, AAMAS '14, Paris, France.

Summit Blue Consulting, L. (2006). Evaluation of the 2005 Energy-Smart Pricing Plan-Final Report.

U.S. Energy Information Administration (2013). Energy Consumption in Homes by End Users. http://www.eia.gov/consumption/residential.

Vytelingum, P., Voice, T. D., Ramchurn, S. D., Rogers, A., and Jennings, N. R. (2010). Agent-based Micro-storage Management for the Smart Grid. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '10, Toronto, Canada. International Foundation for Autonomous Agents and Multiagent Systems.

Wooldridge, M. (2009). *An Introduction to Multiagent Systems*. John Wiley & Sons.

Yang, R. and Newman, M. W. (2013). Learning from a Learning Thermostat: Lessons for Intelligent Systems for the Home. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '13, Zurich, Switzerland. ACM.

Yu, Z. and Dexter, A. (2010). Online Tuning of a Supervisory Fuzzy Controller for Low-energy Building System Using Reinforcement Learning. *Control Engineering Practice*, 18(5):532--539.

2 An Active Learning Approach to Home Heating in the Smart Grid

The content of this chapter has previously appeared in:

> Shann, M. and Seuken, S. (2013). An Active Learning Approach to Home Heating in the Smart Grid. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, IJCAI '13, Beijing, China.

An Active Learning Approach to Home Heating in the Smart Grid

Mike Shann and **Sven Seuken**
Department of Informatics
University of Zurich
{shannm, seuken}@ifi.uzh.ch

Abstract

A key issue for the realization of the smart grid vision is the implementation of effective demand-side management. One possible approach involves exposing dynamic energy prices to end-users. In this paper, we consider a resulting problem on the user's side: how to adaptively heat a home given dynamic prices. The user faces the challenge of having to react to dynamic prices in real time, trading off his comfort with the costs of heating his home to a certain temperature. We propose an active learning approach to adjust the home temperature in a semi-automatic way. Our algorithm learns the user's preferences over time and automatically adjusts the temperature in real-time as prices change. In addition, the algorithm asks the user for feedback once a day. To find the best query time, the algorithm solves an optimal stopping problem. Via simulations, we show that our algorithm learns users' preferences quickly, and that using the expected utility loss as the query criterion outperforms standard approaches from the active learning literature.

1 Introduction

One of society's greatest challenges in the 21st century is the revolution of the energy sector, moving from fossil-based energy sources towards renewable energy like wind and solar. This transition is important to satisfy the growing demand for energy while the annual production of many oil and gas fields is decreasing, and to combat climate change in general and the negative effects of carbon emissions in particular. However, this also creates a number of new challenges for three reasons: energy from renewable sources is very volatile; energy is inherently difficult to store; and the classic model in energy markets is one where supply follows demand. To address these new challenges, governments are investing billions of dollars into the development of the next generation of the electricity grid, the so-called *smart grid* [U. S. Department Of Energy, 2003]. This new electricity network will make it possible to expose real-time prices to end-consumers, use electric vehicles that are plugged into the grid as energy storage devices, and allow power companies to remote control certain home appliances in times when electricity supply is particularly scarce. However, in contrast to the smart grid vision, at the moment most end-users are still facing fixed energy prices or very simple day/night tariffs, and are unaware of changes in the demand or supply of energy.

1.1 Demand-Side Management

With renewable energy becoming a larger part of the overall energy mix, it is becoming increasingly difficult for supply to always follow demand. A number of recent economic and technological studies have shown that effective *demand-side management* will be essential for the success of the smart grid [Cramton and Ockenfels, 2011]. This means that in times where energy supply is scarce, the demand for energy must also decrease. One way to achieve this is to expose dynamic energy prices to end-users in real time such that they can adjust their demand accordingly. At the moment, the biggest demand-response effects come from big companies who already face dynamic prices and can shift some of their energy usage [VDE, 2012]. However, in the future, the percentage of electricity consumed by end-users will increase because more and more cars will be electric vehicles, and an increasing number of homes will use electric heat pumps and air conditioners. Even if just part of the population adopts energy tariffs with dynamic prices, effective demand-response management for end-users will become an important challenge.

1.2 Home Heating with Smart Thermostats

In this paper, we focus on one particular facet of demand-response management: the problem of adaptively heating (and cooling) a user's home given dynamic electricity prices. This addresses an important problem because cooling and heating accounts for the largest part of end-users' energy bills. We consider a future smart grid design, where at least some end-consumers are exposed to dynamic energy prices. To optimize their utility, those users will have to react to dynamic prices in real-time, trading off their comfort (at different temperature levels) with the costs for heating or cooling. Obviously, it is infeasible for a user to always manually change the temperature when a price change occurs. Instead, we envision *smart thermostats* that will automatically reduce the energy consumption of the house when prices are high, but only as much as is justified by the cost savings.

Designing a smart thermostat is a difficult problem because automatically adjusting the temperature requires knowing how the user trades off comfort for money. Some users may have a high value for comfort and may be willing to pay a lot for a perfectly-heated home. Others may be relatively insensitive to temperature changes, and instead would prefer to save on energy costs. Because of this user heterogeneity, the smart thermostat needs to *elicit* the user's preferences and learn this trade-off over time, which makes this a formidable AI problem in the computational sustainability domain.

Yet, even the most sophisticated thermostats currently on the market do not consider this trade-off. The existing devices are able to monitor a home's energy usage and suggest energy saving measures (e.g., *Alert Me*), or they can learn a user's daily schedule and adjust the times at which the house is heated or cooled accordingly (e.g., *Eco Factor* and *Nest*). However, these devices are completely unresponsive to energy price changes. Recent academic work on adaptive home heating has focused on learning the thermal properties of a house, but has also not considered how the user trades off between comfort and money [Rogers *et al.*, 2011]. Naturally, end-consumers are currently still very sceptical regarding the benefits of the smart grid [Jung, 2010]. Many believe that their comfort levels will be reduced and that they will only save little if any money. We argue that a smart thermostat that automatically reacts to price changes is necessary to realize demand-response management, and would also be in the interest of end-users. However, it must be non-intrusive and simple to use, for end-consumers to adopt this technology.

1.3 Overview of Contributions

The main contribution of this paper is an active learning algorithm for the adaptive home heating problem. Our algorithm uses Bayesian inference to learn the user's preferences over time, automatically adjusts the temperature as prices change, and requests new feedback from the user, but only once a day. We explicitly model the user's comfort-cost trade-off by separating the user's value function (for temperature) from the cost function (for heating or cooling). We propose an algorithm that involves solving an optimal stopping problem to find the optimal time to query the user. We evaluate our algorithm in an online fashion via simulations. We find that using the user's expected utility loss as the query criterion outperforms standard approaches from the active learning literature. To the best of our knowledge, we are the first to propose an active learning approach to address demand-side management in the smart grid.

2 Related Work

Automated Control in the Smart Grid. Ramchurn et al. [2012] provide a good introduction to smart grids and the demand-response management challenge. Rogers et al. [2011] study the adaptive home heating problem. However, their focus is on learning the thermal properties of a house and predicting environmental parameters, to optimize the heating schedule. They assume that the user's preferred temperature is known in advance and do not consider the comfort-cost trade-off. McLaughlin et al. [2012] consider the same problem but also assume that the user's desired temperature is known to the algorithm. Vytelingum et al. [2010] study micro-storage management for the smart grid, and devise agent strategies that automatically react to price changes. However, they assume that the amount of energy each user desires per time period is known in advance, and thus the problem of eliciting users' preferences also does not arise in their model. Finally, Jia et al. [2012] consider the retailer's perspective, and provide a solution for optimal pricing of energy, given that users trade off comfort for cost. However, they also do not consider how a demand-response system would learn about a user's trade-off preferences. Overall, our literature review suggests that the problem of *eliciting* and *learning* user preferences in the smart grid has largely been ignored by the research community so far.

Preference Elicitation and Active Learning. Our work primarily uses techniques from preference elicitation [Boutilier, 2002] and active learning [Settles, 2009]. Our Bayesian inference algorithm is inspired by the preference elicitation approach by Chajewska et al. [2000], who use the expected value of information as their query criterion. However, while they consider a domain where arbitrary queries can be synthesized, we consider the problem of selecting the best query from a stream of potential queries which is called *selective sampling* or *stream-based sampling*. Cesa-Bianchi et al. [2006] and Beygelzimer et al. [2009] propose randomized selective sampling algorithms that have good convergence guarantees in the limit, but do not aim to optimize each individual sample. Our query technique is more similar to the approach used by Cohn et al. [1996], in that we aim to minimize the learner's expected error with every individual query. Our work is also related to the label efficient prediction algorithms by Helmbold et al. [1997] and Cesa-Bianchi et al. [2005]. Their algorithms handle the restriction that the learner can only ask a limited number of times, however, they cannot handle context variables, like price for example. In contrast, Krause and Ong [2011] present bandit algorithms that explicitly take context into account. However, they assume that the algorithm receives feedback about the user's utility in every time step which is not given in our domain. Finally, our problem can also be framed as a *partial monitoring game with side information* [Cesa-Bianchi and Lugosi, 2006]. However, existing algorithms for this framework operate in a prior-free domain [Bartók and Szepesvári, 2012], while we assume a Bayesian learning framework.

3 The Model

3.1 Problem Statement

We consider the problem of adaptive home heating over a horizon of N days, where each day consists of K time steps. The price for energy is modeled using a discrete Markov process $\{p_t : t \leq KN\}$. We use T_{out} to denote the current outside temperature, and T to denote the current temperature inside the house. The user's utility is separated into two components. First, it depends on his comfort level, which is mainly determined by the inside temperature T but also influenced by the outside temperature T_{out}. Second, the utility depends on the cost the user has to pay for heating the house, which is a function of the desired inside temperature T, the outside temperature T_{out}, and most importantly the current price for energy p_t. We denote the user's utility by $u(p_t, T, T_{out})$.

Our goal is to design an active learning algorithm that learns the user's preferences over time and controls the house's temperature in a semi-automated way. Every time step, the algorithm receives as input the current price p_t. At most once per day, the algorithm can query the user for the temperature that is currently *optimal* for him:

$$T_{opt}(p_t, T_{out}) = \arg\max_T u(p_t, T, T_{out}). \quad (1)$$

We assume that if the algorithm decides to issue a query, the user provides a temperature value which the algorithm uses to update its model of the user's preferences. Based on its current knowledge, the algorithm then sets the temperature to its current best *estimate* of the optimal temperature, which we denote by $\widehat{T}_{opt}(p_t, T_{out})$.[1] Note that we often use T_{opt} and $\widehat{T}_{opt}$ without the parameters p_t and T_{out} to simplify notation. Our goal is to minimize the user's cumulative utility loss:

$$L = \sum_{t=1}^{KN} \Big(u(p_t, T_{opt}, T_{out}) - u(p_t, \widehat{T}_{opt}, T_{out}) \Big). \quad (2)$$

The one-query-per-day restriction is motivated by our goal of designing a non-intrusive smart thermostat that end-consumers are willing to use. Of course, many other design choices regarding the interaction mode are conceivable, including several queries per day, queries at a fixed time (e.g. in the evening), or even a more intense preference elicitation phase at the beginning of the learning process.

3.2 The User's Utility Function

Inherent to the home heating problem is the user's trade-off between comfort and cost. To model this, we assume a value function $v(T, T_{out})$ that quantifies (in currency) the user's level of comfort for temperature T given T_{out}, and a cost function $c(p_t, T, T_{out})$ that quantifies how expensive it is to heat the house to temperature T at current price p_t given T_{out}. The user's utility is the difference between value and costs:

$$u(p_t, T, T_{out}) = v(T, T_{out}) - c(p_t, T, T_{out}). \quad (3)$$

Value Function. Prior research on thermal comfort has shown that the colder it is outside, the lower the user's acceptable indoor temperature [Peeters *et al.*, 2009]. This suggests that the user's most preferred temperature also depends on the current outside temperature. Formally, we let T^* denote the user's preferred temperature at $T_{out} = 0$, and we let m denote the slope with which the preferred temperature increases as the outside temperature increases. We denote the user's preferred temperature by $T_{pref}(T_{out}) = T^* + mT_{out}$.

Following prior work on home heating (e.g., [Rogers *et al.*, 2011]), we assume that the user incurs a utility loss if the inside temperature deviates from his preferred temperature. In particular, we assume that the utility loss is quadratic in $(T_{pref} - T)$, i.e., in the difference between the user's preferred temperature and the actual inside temperature.

[1] Note that $\widehat{T}_{opt}$ may be different from the temperature value provided by the user, which is consistent with our Bayesian approach, but may be confusing for the user in practice. Of course, to improve usability, the smart thermostat could also "ignore" the Bayesian model for one time step, and simply set the temperature to the value provided by the user.

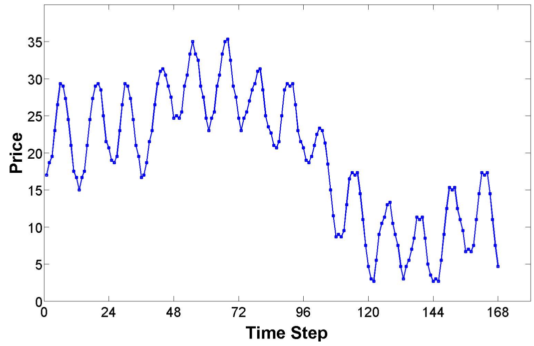


Figure 1: An illustration of the stochastic price process, here over 7 days. The price process has two periodic peaks per day with random fluctuations that follow a random walk.

Peeters et al. [2009] have shown that people are more sensitive to temperature deviations the colder it is outside. To model this, we use an exponential function parameterized by b, which denotes the user's sensitivity if $T_{out} = 0$, and c, which determines how much the user's sensitivity changes as the outside temperature changes. Finally, we let a denote the user's value for his most preferred temperature (i.e., when $T^* + mT_{out} = T$). Putting all of this together, we arrive at the following value function formulation:

$$v(T, T_{out}) = a - \underbrace{b \cdot e^{-c \cdot T_{out}}}_{\text{sensitivity}} \Big(\underbrace{(T^* + mT_{out})}_{\text{preferred temp.}} - T \Big)^2 \quad (4)$$

Cost Function. The user's cost function is given by the following equation:

$$c(p, T, T_{out}) = p|T - T_{out}|. \quad (5)$$

This function captures the fact that the flow of heat between a building's interior and exterior is proportional to the temperature difference, which implies that the amount of energy necessary to heat a house also depends on the temperature difference. Note that this function correctly models heating and cooling, since it only depends on the temperature difference.

Combining the value and the cost function, we obtain the following *linearly separable utility function*:

$$u(p, T, T_{out}) = \underbrace{a - be^{-cT_{out}} \big((T^* + mT_{out}) - T\big)^2}_{\text{value}} - \underbrace{p|T - T_{out}|}_{\text{cost}}$$

3.3 The Stochastic Price Process

Because the algorithm only queries the user once per day, we are interested in the daily price dynamics. An important feature of the daily energy prices are two peaks, one in the morning at around 8 a.m., and one in the evening at around 6 p.m. We model this periodicity using a sine function, following Weron [2006]. To model any random price movements (e.g., due to demand or supply changes) we use a discrete symmetric random walk. Put together, the price process is given by:

$$p_t = A \sin(\omega t + \phi) + B + p_{t-1} + X_t, \quad (6)$$

where A is the amplitude of the sine, ω is the periodicity, ϕ is the phase shift, B is the offset, and X_t is a Bernoulli variable corresponding to the random walk. We use the notation $\mathrm{P}(p_{t'}|p_t)$ to denote the conditional probability of encountering the price $p_{t'}$ given p_t. See Figure 1 for an illustration of the price process over 7 days with 24 time steps per day.

4 The Active Learning Algorithm

The active learning algorithm we propose consists of two main components: 1) a Bayesian learning component that learns the parameters of the user's utility function over time, and 2) a query component that decides when to ask the user for new feedback (once per day). In Section 4.1, we describe the high-level algorithmic framework, before diving into the details of the two components in the following sections.

4.1 The Algorithmic Framework

Every day, the algorithm's goal is to select the best query from the stream of prices it encounters. Loosely speaking, it faces the following gamble. Either sample at the current price or wait and hope that a future query will yield a more useful sample. We will use the notion of a *gain function* $G(p_t)$ to measure the "usefulness" of a query at price p_t. It is intuitive, for example, that querying at a price at which the user has already given feedback before is less useful than asking at a price that has not been encountered before. The different gain functions we consider (information gain and variance reduction) measure usefulness in different ways and thus lead to different decisions regarding the optimal query time.

Given the K times steps per day, the algorithm's goal is to find the optimal stopping time $t^* \in \{1, \ldots, K\}$ at which the expected gain G of a query is highest:

$$t^* = \arg\max_t \mathrm{E}[\mathrm{G}(p_t)]. \tag{7}$$

To find the optimal stopping time, the algorithm computes an *optimal stopping policy* $\pi(t, p_t) \rightarrow \{sample, continue\}$. For each time t and price p_t, this policy prescribes whether to ask the user for feedback now, or whether to wait. This policy can be computed by dynamic programming [Peskir and Shiryaev, 2006]. Keep in mind that the algorithm computes a new optimal stopping policy at the beginning of every day.

If the algorithm decides to request feedback, it asks the user what his preferred temperature is right now given p_t and T_{out}. The user decides how to trade off his comfort level against the costs for heating, and then provides a temperature value y_t to the algorithm. Using this new data point, the algorithm updates its model of the user's utility function using Bayes' rule. Finally, the algorithm sets the optimal temperature, $\widehat{T}_{opt}$, taking into account its prior knowledge and all feedback data it has gathered about the user's preferences so far. The user then suffers a utility loss of $u(p_t, T_{opt}, T_{out}) - u(p_t, \widehat{T}_{opt}, T_{out})$, which is not observed by the algorithm, but which we use to measure the performance in our simulation in Section 5. The whole active learning framework is shown in Algorithm 1.

4.2 Bayesian Updating & Setting the Temperature

Recall that the user's optimal temperature is given by:

$$T_{opt}(p_t, T_{out}) = \arg\max_T u(p_t, T, T_{out}). \tag{8}$$

Based on the functional form of the user's utility function described in Section 3.2, we can calculate the first order condition and solve for T, and arrive at the following equation for the user's optimal temperature:

$$T_{opt}(p, T_{out}) = T^* + mT_{out} \pm p\frac{e^{cT_{out}}}{2b} \tag{9}$$

Algorithm 1: Active Learning Framework

Input: prior $(m_{\boldsymbol{\theta}}, \Sigma_{\boldsymbol{\theta}})$; noise variance σ_n^2
Variables: current price p_t, optimal stopping policy π
begin
  for *d=1 to # of days* **do**
    for *t=1 to # of time steps per day* **do**
      $p_t \leftarrow getNextPrice(p_{t-1})$
      if *t =1* **then**
        //allow a new query
        canAsk ← **true**
        $\pi \leftarrow OptimalStopping(p_t)$
      if *canAsk* **then**
        //decide whether to query user
        if $\pi(p_t, t) = sample$ **then**
          $y_t \leftarrow getUserFeedback()$
          $BayesianUpdate(p_t, y_t)$
          canAsk ← **false**
      $\widehat{T}_{opt} \leftarrow SetTemperature(p_t, T_{out})$

Note that a does not matter for the optimization, and we only have to learn the parameter vector $\boldsymbol{\theta} = (b, c, T^*, m)$.

Bayesian Updating. We assume that the parameter vector $\boldsymbol{\theta}$ is normally distributed, and therefore define a Gaussian prior $\mathrm{P}(\boldsymbol{\theta}) = \mathcal{N}(m_{\boldsymbol{\theta}}, \Sigma_{\boldsymbol{\theta}})$. Furthermore, we assume that the user makes mistakes when giving feedback y_t to the thermostat. We model this with i.i.d. additive Gaussian noise, $y_t = T_{opt} + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$ is a normally distributed random variable with mean 0 and *noise variance* σ_n^2. Thus, the likelihood of y_t is also normally distributed with mean T_{opt} and variance σ_n^2:

$$\mathrm{P}(y_t|p_t, \boldsymbol{\theta}) \propto \exp\Big(-\frac{1}{2\sigma_n^2}(y_t - T_{opt})^2\Big). \tag{10}$$

The posterior is then computed as the product of the prior and the likelihood according to Bayes' rule:

$$\mathrm{P}(\boldsymbol{\theta}|p_t, y_t) \propto \mathrm{P}(\boldsymbol{\theta}) \cdot \mathrm{P}(y_t|p_t, \boldsymbol{\theta}). \tag{11}$$

To update the posterior distribution after a sample point (p_t, y_t) has been gathered, we use Eq. (11) recursively, using the posterior after $k{-}1$ observations as the prior for the k^{th} update step:

$$\mathrm{P}(\boldsymbol{\theta}|D_{k-1} \cup (p_t, y_t)) \propto \mathrm{P}(\boldsymbol{\theta}|D_{k-1}) \cdot \mathrm{P}(y_t|p_t, \boldsymbol{\theta}), \tag{12}$$

where $D_{k-1} = \{(p_{i_1}, y_{i_1}), ..., (p_{i_{k-1}}, y_{i_{k-1}})\}$ denotes all $k{-}1$ data points the algorithm has gathered until time step $t-1$.

Setting the Temperature. Finally, the thermostat sets the estimated optimal temperature (according to its model of the user's preferences) by computing the expected value of T_{opt}, weighting each of the possible values for the parameters $\boldsymbol{\theta}$ by their posterior probability:

$$\widehat{T}_{opt}(p_t, T_{out}) = \mathbb{E}_{\boldsymbol{\theta}}[T_{opt}] \tag{13}$$

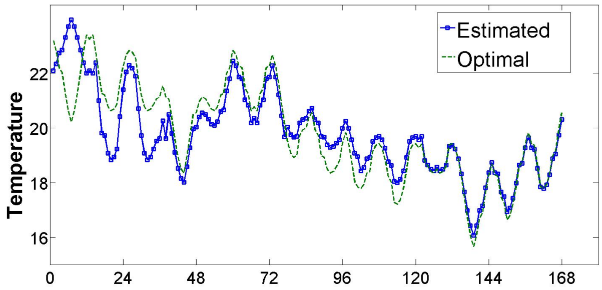


Figure 2: A sample run, illustrating how our algorithm learns the user's preferences over time (here 7 days).

Figure 2 illustrates what our algorithm does in practice. The figure shows a sample run from our simulation (described below), over 7 days, here with 24 time steps per day. The blue line represents the user's true optimal temperature. The green line represents the estimated temperature values that our algorithm sets based on its user model. As one can see, although the estimated temperature is initially off by 2 to 3 degrees, it quickly converges to the true optimal temperature.

4.3 Optimal Stopping using Information Gain

Now that we have introduced the learning component of our algorithm, we move on to the description of the query component. First, we formalize the optimal stopping problem and show how to solve it. Then we introduce *information gain* as the first gain function, or query criterion. In the next two sections, we refine those initial approaches, leading to an improved version of the optimal stopping algorithm as well as to more sophisticated query criteria.

Computing the Optimal Stopping Policy. Recall from Section 4.1 that the optimal stopping policy is a function $\pi(t, p)$ that for every price p and time step t prescribes whether to query now, or whether to wait. Obviously, the policy only prescribes to *wait* if the immediate gain from querying the user now is lower than the expected future gain from waiting and querying later.

While $G(p_t)$ denotes the immediate gain from querying now at price p_t, we let S_t denote the expected gain at time step t when following the optimal stopping policy at every time step going forward from t. S_t is defined recursively as:

$$S_t = G(p_t) \quad \text{for } t = K \quad \text{(last time step)}$$
$$S_t = \max\{G(p_t), \mathbb{E}[S_{t+1}|p_t]\} \quad \text{for } t = K{-}1, \dots, 1. \quad (14)$$

To derive the optimal policy, we compare the gains $G(p_t)$ at time step $t = 1, \dots, K-1$ to the expected future gains $\mathbb{E}[S_{t+1}|p_t]$ for all possible prices p_t. If $G(p_t) \geq \mathbb{E}[S_{t+1}|p_t]$, then the optimal policy states that we should query, i.e., $\pi(t, p_t) = sample$. Otherwise, $\pi(t, p_t) = continue$.

Note that the first price p_1 is known, and thus all future prices p_t that could possibly be encountered until the end of the day can be computed by adding or subtracting a) the random walk price increment per time step, and b) the price movements according to the daily price process model.

Algorithm 2: Computing the Optimal Stopping Policy

Input: starting price p_1
Output: optimal stopping policy π
begin
    $S \leftarrow 0$
    for t = *# of time steps per day to 1* **do**
        forall the *reachable prices* p **do**
            if *t = # of time steps per day* **then**
                $\pi(t,p) = sample$
            else
                if *t = # of time steps per day* -1 **then**
                    $S_{t,p} \leftarrow \frac{1}{2}[\mathrm{G}(p+1) + \mathrm{G}(p-1)]$
                else
                    $S_{t,p} \leftarrow \frac{1}{2}[\max\{\mathrm{G}(p+1), S_{t+1,p+1}\} + \max\{\mathrm{G}(p-1), S_{t+1,p-1}\}]$
                if $G(p) \geq S_{t,p}$ **then**
                    $\pi(t,p) = sample$
                else
                    $\pi(t,p) = continue$
    return π

Algorithm 2 shows how the optimal stopping policy is computed for all time steps and all possible prices. To simplify the exposition of the algorithm, we assume here that the price process is a symmetric random walk with step size 1. However, it is straightforward to adopt the algorithm to more complicated price processes such as the one defined in Section 3.3. We use the variable $S_{t,p}$ to denote the expected gain at time step t, given price p, i.e. $S_{t,p} = \mathrm{E}[S_t|p]$.

Query criterion: Information Gain. So far, we have left the gain function $G(p_t)$ unspecified. However, to instantiate the optimal stopping algorithm, we need to define one particular gain function, or query criterion, $\mathrm{G}(p_t)$, that quantifies how useful a query is at a price p_t (note that we use the terms *gain function* and *query criterion* interchangeably). The first criterion we discuss is *information gain* which measures how much the uncertainty about the parameters $\boldsymbol{\theta}$ is reduced by adding an observation y_t [Cover and Thomas, 2006]. This is expressed using the mutual information $I(\theta, y_t) = H(\theta) - H(\theta|y_t)$, where $H(\cdot)$ is the differential entropy [Cover and Thomas, 2006]. Intuitively, the higher the uncertainty (or variance) of T_{opt} at a given price, the more information can be gathered by querying at this price. It can be shown that the information gain for a given price is equivalent to the variance of the predicted optimal temperature T_{opt} [MacKay, 1992]. Thus, we define our first query criterion as:

$$\mathrm{G}^{inf}(p_t) = Var[T_{opt}(p_t)] \quad (15)$$

Note that the user's utility actually also depends on the outside temperature T_{out}. However, in this paper, we do not assume that the algorithm has a model for T_{out}. Thus, our formulation of the optimal stopping problem is only optimal with respect to the stochastic price process and implicitly assumes a fixed value for T_{out}. But it is straightforward to extend the algorithm by incorporating a model for T_{out} as well.

4.4 Optimal Stopping using Temperature Loss

Note that the basic version of the optimal stopping algorithm neglects the fact that until the algorithm asks the user for feedback, the user has already incurred a utility loss every time step. Therefore, we now re-formulate the optimal stopping problem using *loss functions*, with the new goal of minimizing the expected total loss. Therefore, we define our gain function $G(p_t)$ to be a loss function multiplied by -1, i.e., $G(p_t) = -L(p_t)$, such that minimizing the expected loss is equivalent to maximizing the expected gain.

We define the function $L^{now}(p_t)$ that measures the loss the user incurs at time t given price p_t if the algorithm estimates the optimal temperature with its current knowledge without issuing a query. Thus, the algorithm will incur loss $L^{now}(p_t)$ at every time step t until it decides to query the user. However, if the algorithm decides to issue a query at time t, then the loss incurred will be smaller than $L^{now}(p_t)$ because the algorithm will be able to estimate the temperature more accurately due to one additional data point. This leads to the following new definition of S_t:

$$S_t = G(p_t) \quad \text{for } t = K \quad \text{(last time step)}$$
$$S_t = \max\{G(p_t), -L^{now}(p_t) + \mathrm{E}[S_{t+1}|p_t]\} \ \text{ for } t = K\text{–}1,\ldots,1.$$

The term $-L^{now}(p_t)$ in the last equation reflects the loss that the user incurs if the algorithm does not issue a query at time t, while the (smaller) loss incurred if the algorithm issues a query will be incorporated in the gain function $G(p_t)$, which we define in the next section. As before, the optimal stopping policy can be computed using the approach summarized in Algorithm 2, but adapting the equations for the expected future gains $S_{p,t}$ according to the new formulation.

Query criterion: Temperature Loss. To instantiate the new *loss-based* optimal stopping algorithm, we follow an idea from [Cohn *et al.*, 1996], and specify as our new goal to select the query that minimizes the expected squared error in the temperature estimation. This is motivated by the fact that the expected squared error of a learner can be decomposed into squared bias and variance, the so-called *bias-variance decomposition* [Geman *et al.*, 1992], which states that we can approximate the expected squared predictive error if the bias of the learner is sufficiently small compared to the variance.

First, let us revisit $L^{now}(p_t)$. Due to the bias-variance decomposition, we can approximate this function using the variance of the predicted temperature: $L^{now}(p_t) = Var[T_{opt}(p_t)]$. To obtain a gain function $G(p_t)$, we need the expected (posterior) variance of T_{opt}, condition on sampling at a given price p_t. We let $L^{ask,t}_{temp}(p)$ denote the expected conditional variance of T_{opt} at price p, if the user was queried at time step t, i.e. $L^{ask,t}_{temp}(p) = Var[T_{opt}(p)|(p_t, y_t)]$. The gain function that we define now amounts to quantifying the expected predictive loss until the end of the day plus the expected loss of one additional day, given the user was queried. Adding the expected loss of one additional day is a heuristic to account for the future differences in losses due to the particular query. This is only a heuristic as it does not account for *all* effects on losses in future days, because it ignores the fact that the algorithm will be able to issue a new query on the next day (and on every day thereafter).[2] The query criterion is then defined as follows:[3]

$$G^{loss\_temp}(p_t) =$$
$$-\Big(L^{ask,t}_{temp}(p_t) + \underbrace{\sum_{t'=t+1}^{K} \mathrm{E}[L^{ask,t}_{temp}(p_{t'})|p_t]}_{\text{loss until end of day}} + \underbrace{\sum_{t'=K+1}^{2K} \mathrm{E}[L^{ask,t}_{temp}(p_{t'})|p_t]}_{\text{loss next day}}\Big)$$

Note that $\mathrm{E}[L^{ask,t}_{temp}(p_{t+i})|p_t]$ denotes the expectation of $L^{ask,t}_{temp}(p_{t+i})$ with respect to the condition probability distribution given by the price process, i.e., according to $\mathrm{P}(p_{t+i}|p_t)$, as defined in Section 3.3.

4.5 Optimal Stopping using Utility Loss

The query criterion we develop in this section is based on the following insight: minimizing the expected squared error of the temperature estimation (as we did in the previous section) misses the fact that the user primarily cares about his *utility losses*, and that an error in the temperature estimation can only be a proxy for that. Thus, our new goal is to directly minimize the user's expected utility loss.

Analogously to the temperature variance criterion, we can approximate the user's squared utility loss with the variance of the utility function. To arrive at the expected utility loss we can simply take the square root of the variance of the utility. Therefore, define $L^{now}_u(p) = \sqrt{Var[u(p)]}$, and similarly $L^{ask,t}_u(p) = \sqrt{Var[u(p)|(p_t, y_t)]}$. The following query criterion minimizes the expected square root of the variance of the utility function, which is equivalent to choosing a sample point that minimizes the user's expected utility loss:

$$G^{loss\_util}(p_t) =$$
$$-\Big(L^{ask,t}_{u}(p_t) + \underbrace{\sum_{t'=t+1}^{K} \mathrm{E}[L^{ask,t}_{u}(p_{t'})|p_t]}_{\text{loss until end of day}} + \underbrace{\sum_{t'=K+1}^{2K} \mathrm{E}[L^{ask,t}_{u}(p_{t'})|p_t]}_{\text{loss next day}}\Big)$$

This query criterion together with the optimal stopping formulation described above is the ultimate query component that we propose for our active learning algorithm.

5 Experiments

We evaluate our active learning approach via simulations, following the basic structure of Algorithm 1. For the learning and prediction part of the algorithm, we perform a non-linear regression using a Bayesian linear parameter model.

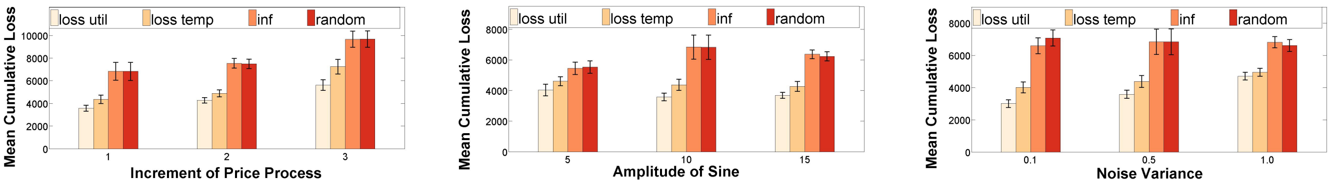
5.1 Bayesian Linear Parameter Model

Recall that the optimal temperature is a non-linear function of the input variables p_t and T_{out}. However, if we fix the parameter c, we can write the optimal temperature as a linear parameter model:

$$T_{opt}(p_t, T_{out}) = w_0 + w_1 T_{out} + w_2 p \cdot e^{cT_{out}}/2 \qquad (16)$$

[2]Note that solving an optimal stopping problem over a horizon of N days with K time steps each quickly becomes computationally infeasible, even for moderate values of N and K.

[3]To simplify the notation for the summation indices, we only state the criterion for the first day.



(a) (b) (c)

Figure 3: Simulation results comparing four different query criteria: (a) varying the increment of the price process; (b) varying the amplitude of the sine of the price process; (c) varying the noise variance σ_n^2.

We can identify the weights as follows: $w_0 = T^*$, $w_1 = m$ and $w_2 = 1/b$. We augment the input vector with an offset, such that $\mathbf{x} = (1, p_t, T_{out})$ and write

$$T_{opt}(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}), \tag{17}$$

where $\mathbf{w}=(w_0, w_1, w_2)^T$ and $\boldsymbol{\phi}(\mathbf{x})=(1, T_{out}, pe^{cT_{out}}/2)^T$. Due to our assumption of a Gaussian prior and a Gaussian additive noise model, the posterior probabilities are likewise Gaussian and we can perform Bayesian regression using the Bayesian linear parameter model [Bishop, 2006].

5.2 Experimental Set-up

For all experiments, we use the following basic set-up. We use $N = 30$ days, each day consisting of $K = 12$ time steps. The prior means are 22 for w_0 (i.e. T^*), 0.1 for w_1 (i.e. m), and 0.2 for w_2 (i.e., $1/b$). The values $T^* = 22$ and $m = 0.1$ are similar to the values reported by Peeters et al. [2009]. The prior variances are fixed as $\boldsymbol{\sigma^2}=(1, 0.1, 0.1)$. The noise variance, which describes the user's ability to provide accurate temperature values (see also Eq. (10)), is set to $\sigma_n^2 = 0.5$.

For the sine of the price process, we set the amplitude $A = 10$, the offset $B = 20$, the periodicity $\omega = 4\pi/K$, and the phase shift $\phi = 4\pi/3$. The increment of the random walk is 1, i.e. $X_t \in \{-1, 1\}$. The daily variations of the outside temperature are modeled using a sine function with offset 5 and amplitude 5. Thus, T_{out} ranges from 0 to 10 degrees during a day, which are typical heating conditions [Peeters *et al.*, 2009]. The parameter c is set to 0.01. We also conducted the simulations with higher values of c but found qualitatively similar results. Each experiment is repeated for 100 trials, and in every trial, a user type is drawn randomly from the Gaussian prior distribution.

5.3 Results

We compare the performance of the following four query criteria: (1) G^{inf}, (2) G^{loss_temp}, (3) G^{loss_util}, and (4) random querying. All four query criteria are run in parallel, which implies that they see the same price process and even get the same samples if they perform a query at the same time step.

We vary the parameters that we identified to have a significant impact on the performance of the query criteria. Figure 3(a) shows the results of increasing the increment of the random walk, X_t, from 1 to 2 to 3. As one can see, the query criterion G^{loss_util} performs significantly better than all other criteria, for small as well as for large price increments. In Figure 3(b), we present performance results varying the amplitude of the sine of the prices process from 5 to 10 to 15. Again, G^{loss_util} outperforms all other query criteria for all three settings. Lastly, in Figure 3(c), we vary the noise variance, σ_n^2, from 0.1 to 0.5 and 1.0. Here, G^{loss_util} performs significantly better than all query criteria for $\sigma_n^2 = 0.1$ and σ_n^2 it performs equally well as G^{loss_temp} for $\sigma_n^2 = 1.0$. In summary, G^{loss_util} is never worse than the other criteria, and in most settings significantly outperforms all other criteria.

The results also demonstrate that the information gain criterion, i.e., G^{inf}, performs much worse than G^{loss_temp} and G^{loss_util}. This is mainly due to the fact that the latter two criteria take the loss over the whole day into account, whereas information gain neglects this. A second finding is that the larger the noise, the smaller the differences between the individual criteria. This also makes sense, because lots of noise decreases the predictability of the queries which decreases the value of sophisticated optimized techniques.

6 Conclusion

In this paper, we have studied the problem of adaptively heating a home given dynamic energy prices. We have presented a novel active learning algorithm that determines the optimal time to query the user for feedback, learns the user's preferences via Bayesian updating, and automatically sets the temperature on the user's behalf as prices change. Given the constraint of at most one query per day, determining the optimal query time requires solving an optimal stopping problem. Via simulations, we have demonstrated that a query criterion that minimizes the user's expected utility loss outperforms standard approaches from the active learning literature.

It is important to note that we have purposefully presented a relatively simple user model and made a number of simplifying assumptions that we will relax in future work. As a first step, we plan on incorporating the temporal dynamics of heating as well as weather forecasts into our model. This will give rise to a sequential planning problem, which we can combine with our active learning algorithm.

We believe that AI techniques such as preference elicitation and active learning are essential to mediate the interactions between end-consumers and the energy market. To realize the smart grid vision of the future, the design of suitable user interfaces and the use of learning algorithms may ultimately prove to be as important as the economic design of the energy market or the technical aspects of the smart grid.

Acknowledgments

We would like to thank Timo Mennle and Siddhartha Ghosh for insightful discussions and the anonymous reviewers for their helpful comments.

References

[Bartók and Szepesvári, 2012] Gábor Bartók and Csaba Szepesvári. Partial Monitoring with Side Information. In *Proceedings of Algorithmic Learning Theory (ALT)*, 2012.

[Beygelzimer *et al.*, 2009] Alina Beygelzimer, Sanjoy Dasgupta, and John Langford. Importance Weighted Active Learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2009.

[Bishop, 2006] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[Boutilier, 2002] C. Boutilier. A POMDP Formulation of Preference Alicitation Problems. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2002.

[Cesa-Bianchi and Lugosi, 2006] Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[Cesa-Bianchi *et al.*, 2005] Nicolò Cesa-Bianchi, Gábor Lugosi, and Gilles Stoltz. Minimizing Regret with Label Efficient Prediction. *IEEE Transactions of Information Theory*, 51(6):2152–2162, 2005.

[Cesa-Bianchi *et al.*, 2006] Nicolò Cesa-Bianchi, Claudio Gentile, and Luca Zaniboni. Worst-case Analysis of Selective Sampling for Linear Classification. *Journal of Machine Learning Research*, 7:1205–1230, 2006.

[Chajewska *et al.*, 2000] U. Chajewska, D. Koller, and R. Parr. Making Rational Decisions Using Adaptive Utility Elicitation. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2000.

[Cohn *et al.*, 1996] David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. Active Learning with Statistical Models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.

[Cover and Thomas, 2006] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (2. ed.)*. Wiley, 2006.

[Cramton and Ockenfels, 2011] Peter Cramton and Axel Ockenfels. Economics and Design of Capacity Markets for the Power Sector. Working Paper, University of Maryland, May 2011.

[Geman *et al.*, 1992] Stuart Geman, Elie Bienenstock, and René Doursat. Neural Networks and the Bias/Variance Dilemma. *Neural Computation*, 4(1):1–58, January 1992.

[Helmbold and Panizza, 1997] David P. Helmbold and Sandra Panizza. Some Label Efficient Learning Results. In *Proceedings of the Conference on Learning Theory (COLT)*, 1997.

[Jia and Tong, 2012] Liyan Jia and Lang Tong. Optimal Pricing for Residential Demand Response: A Stochastic Optimization Approach. In *Proceedings of the Allerton Conference on Communication, Control and Computing*, 2012.

[Jung, 2010] Alexander Jung. Teure Ersparnis. http://www.spiegel.de/spiegel/0,1518,711967,00.html. Accessed: October 2010.

[Krause and Ong, 2011] Andreas Krause and Cheng Soon Ong. Contextual Gaussian Process Bandit Optimization. In *Proceedings Neural Information Processing Systems (NIPS)*, 2011.

[MacKay, 1992] David J. C. MacKay. Information-based Objective Functions for Active Data Selection. *Neural Computation*, 4(4):590–604, 1992.

[McLaughlin *et al.*, 2012] Zhe Yu Linda McLaughlin, Liyan Jia, Mary C. Murphy-Hoye, Annabelle Pratt, and Lang Tong. Modeling and Stochastic Control for Home Energy Management. In *Proceedings of the Power and Energy Society (PES) General Meeting*, 2012.

[Peeters *et al.*, 2009] Leen Peeters, Richard de Dear, Jan Hensen, and D'haeseleer William. Thermal Comfort in Residential Buildings: Comfort Values and Scales for Building Energy Simulation. *Applied Energy*, 86(5):772–780, 2009.

[Peskir and Shiryaev, 2006] Goran Peskir and Albert Shiryaev. *Optimal Stopping and Free-Boundary Problems*. Lectures in Mathematics, ETH Zürich. Birkhäuser Verlag, 2006.

[Ramchurn *et al.*, 2012] Sarvapali Ramchurn, Perukrishnen Vytelingum, Alex Rogers, and Nicholas R. Jennings. Putting the "Smarts" into the Smart Grid: A Grand Challenge for Artificial Intelligence. *Communications of the ACM*, 55(4):86–97, 2012.

[Rogers *et al.*, 2011] Alex Rogers, Sasan Maleki, Siddhartha Ghosh, and Jennings Nicholas R. Adaptive Home Heating Control Through Gaussian Process Prediction and Mathematical Programming. In *Proceedings of the Second International Workshop on Agent Technology for Energy Systems (ATES)*, 2011.

[Settles, 2009] B. Settles. Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[U. S. Department Of Energy, 2003] U. S. Department Of Energy. Grid 2030: A National Vision For Electricity – Second 100 Years. 2003.

[VDE, 2012] VDE. Demand Side Integration – Lastverschiebungspotenziale in Deutschland. Technical Report, VDE Verband the Elektrotechnik, June 2012.

[Vytelingum *et al.*, 2010] Perukrishnen Vytelingum, Thomas D. Voice, Sarvapali D. Ramchurn, Alex Rogers, and Nicholas R. Jennings. Agent-based Micro-storage Management for the Smart Grid. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2010.

[Weron, 2006] Rafal Weron. *Modeling and Forecasting Electricity Loads and Prices: A Statistical Approach*. HSC Books. Hugo Steinhaus Center, Wroclaw University of Technology, 2006.

3 Adaptive Home Heating under Weather and Price Uncertainty using GPs and MDPs

The content of this chapter has previously appeared in:

> Shann, M. and Seuken, S. (2014). Adaptive Home Heating under Weather and Price Uncertainty using GPs and MDPs. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '14, Paris, France.

Adaptive Home Heating under Weather and Price Uncertainty using GPs and MDPs

Mike Shann and Sven Seuken
Department of Informatics
University of Zurich
{shannm, seuken}@ifi.uzh.ch

ABSTRACT

We consider the problem of adaptive home heating in the smart grid, assuming that real-time electricity prices are being exposed to end-users with the goal of realizing demand-side management. To lower the burden on the end-users, our goal is the design of a *smart thermostat* that automatically heats the home, optimally trading off the user's comfort and cost. This is a challenging problem due to two sources of uncertainty: future weather conditions and future electricity prices. Our main technical contribution is a general technique that uses predictive distributions obtained from Gaussian Process (GP) regressions to compute the state transition probabilities of an MDP, such that the solution to the resulting MDP constitutes a sequentially optimal policy. We apply this general approach to the home-heating problem, where we use the predictive distributions of the GPs for the day-ahead external temperatures and electricity prices. The solution to the home-heating MDP constitutes an optimal heating policy that maximizes the user's utility given the probability information gathered by the Gaussian process model. Via simulations we show that our MDP-based approach outperforms various benchmarks, especially for cost-sensitive users.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Plan execution, formation, and generation*

Keywords

Smart Grid; Home Heating; Real-time Prices; MDPs; GPs

1. INTRODUCTION

The electricity grid is undergoing big changes as many countries are now moving from fossil fuel burning power stations to renewable energies (solar, wind, tidal). This creates a number of challenges because energy from renewable sources is very volatile, energy is inherently difficult to store, and the classic model in energy markets is one where supply follows demand. Until now, end-users have generally faced fixed energy prices and were not aware of changes in supply and demand of energy. But with more and more renewable energy sources, this inelastic demand side becomes an increasingly severe problem [2]. For this reason, governments around the world are investing billions in the development of the next generation of the electricity grid, the so-called *smart grid*.

One important part of the smart grid vision is to create a paradigm shift that enables *demand-side management*. This means that in times when energy is scarce (and expensive), the demand for energy should adjust and go down, and when energy is plentiful (and cheap), the demand for energy should go up. One way to achieve demand-side management is by exposing real-time prices to end-users. While currently the biggest potential for demand-side management still lies in the industrial sector, this will change very soon, with more and more people driving electric vehicles and heating their homes with electricity instead of oil or gas.

1.1 Home Heating in the Smart Grid

The energy used for heating homes is a major part of many countries' energy consumption and consequently also accounts for a substantial part of their CO_2 emissions. In the US, approximately 40% of household energy is used for heating [15], and in the UK it even accounts for 66% of household energy usage [7]. Thus, if the international community wants to meet its goal of reducing CO_2 emissions as stated in the Kyoto protocol [14], reducing the energy used for heating must be part of the agenda. Of course, individual home owners also have an interest in this, given that home heating accounts for the majority of their household energy costs. Thus, the optimization of energy usage for home heating is an important lever to reduce CO_2 emissions, to enable demand-side management, and to reduce individual home owners' energy costs.

There are two main avenues for improving the energy efficiency of homes. One is better insulations, which reduces the leakage of heat to the outside. However, this is often very expensive or not even worth it, especially for old buildings. The other avenue, which we consider here, is the optimization of the home heating control process. We assume that the heating device is a heat pump that works with electricity (an assumption that will be true for many households once renewable energy makes up the majority of the energy mix). Heat pumps offer the advantage of higher energy efficiency and lower CO_2 emissions compared to conventional forms of heating, in particular when renewable energy sources are used to produce the electricity. When heat pumps are powered by electricity, home heating is obviously directly connected to the electricity market. Thus, for demand-side management to be effective, the home heating controller must be responsive to price changes, which adds a new complication to this problem.

1.2 Coping with Weather & Price Uncertainty

Our goal is to design a *smart thermostat* that has a model of the user's preferences and automatically adjusts the temperature as the environmental conditions affecting the heating (such as external temperature and electricity prices) change. To optimize the heating

Appears in: *Alessio Lomuscio, Paul Scerri, Ana Bazzan, and Michael Huhns (eds.), Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014), May 5-9, 2014, Paris, France.*

strategy, the smart thermostat must "plan ahead," e.g., if electricity prices are about to rise, then the current cheap electricity should be used to heat up the house, such that it is already warm during times of high energy costs. Therefore, the smart thermostat first needs to predict the future development of the electricity price and the external temperature, and then use this information to compute a heating strategy that is optimal for the user.

Our work is motivated by earlier research on home heating by Rogers et al. [10], who developed an adaptive heating algorithm that first predicts future external temperatures using Gaussian Processes (GPs), and then computes a heating plan using mixed-integer programming. However, their algorithm implicitly assumes that the weather predictions are correct. Consequently, the home heating policy they compute might be sub-optimal because it does not account for the uncertainty inherent to weather forecasts. This issue is exacerbated when prices are dynamic and therefore not known perfectly in advance. In our approach, we also use GPs to predict future outside temperatures, as well as future electricity prices. However, we use Markov Decision Processes (MDPs) to explicitly account for the uncertainty of these predictions.

1.3 Overview of Contributions

Our main technical contribution in this paper is a general technique that uses the probabilistic predictions obtained from Gaussian Process regressions to define the state transitions for an MDP, such that a solution to the resulting MDP constitutes a sequentially optimal policy for the problem. This approach can be applied to any problem that requires a stochastic policy that is contingent on the future values of certain state variables.

We illustrate this general technique by applying it to the problem of computing a sequentially optimal home heating policy. Using the predictive distributions of the GPs for the day-ahead external temperature and electricity prices, the solution to the MDP constitutes a heating policy that maximizes the user's total utility. Our MDP formulation is very general, and can easily be extended to incorporate other sources of uncertainty (e.g., home occupancy).

We use simulations based on real-world weather data to compare our MDP-based algorithm against multiple benchmarks from the literature (including MIPs and MPCs). We demonstrate that our approach achieves the same or higher performance, and is particularly effective for cost-sensitive users.

2. RELATED WORK

Rogers et al. [11] provide an introduction to the smart grid from a multi-agent systems perspective, and Ramchurn et al. [8] describe the opportunities for AI research in this field. Vytelingum et al. [16] study autonomous agents for micro-storage in the smart grid that automatically react to price changes. However, their approach does not explicitly account for the uncertainty in the domain.

In our own prior work on home heating, we have studied how to automatically learn the user's preferences (trading off comfort with costs) with minimal interactions [13]. In this paper, we assume that the thermostat already has a good model of the user's preferences, and focus on computing a sequentially optimal heating policy. However, our MDP-based approach naturally lends itself towards incorporating preference elicitation techniques as presented in our prior work, which is subject to our ongoing research.

Various researchers have studied the problem of energy efficient heating control. A notable approach involves predicting the occupancy of the building with the goal of reducing the inside temperature when the building is unoccupied. For example, Scott et al. [12] use motion sensing, and find patterns in user behavior to heat adaptively. Occupancy prediction is complementary to weather and price prediction, but our MDP-based approach can easily be extended to also include an occupancy prediction component.

In the control community, the state-of-the-art method for home heating is *model predictive control (MPC)*. For example, *OptiControl* is a project aiming at energy efficient heating of office buildings [5]. They consider weather forecasts and occupancy predictions, and use MPCs to compute a heating policy. A similar approach is followed by Yu et al. [4]. MDPs and MPCs share some commonalities, but there are also important differences. In Section 6.2, we provide a detailed comparison of the two methods.

3. THE MODEL

We consider the problem of computing a sequentially optimal home heating policy that reacts to changing environmental conditions.[1] We discretize every day into T intervals, each consisting of $\Delta t = 24h/T$ (a typical interval length is 10 minutes). Each time step, we consider three environmental variables: the internal temperature T_t^{int}, the external temperature T_t^{ext}, and the price of electricity p_t. The heat pump is controlled via a decision variable h_t. Depending on how well the heat pump can be controlled, this variable is either binary, $h_t \in \{0, 1\}$, corresponding to the heating being turned off or on; or the variable is continuous, $h_t \in [0, 1]$, corresponding to the pump operating at a certain level between zero and maximum power. To compute the optimal heating policy, we need the following four components:

1. A thermal model of the house,
2. a model of the user's preferences,
3. a prediction of future environmental conditions, and
4. an optimization method that, given the thermal model and the predictive information, computes an optimal heating plan according to some criterion of optimality.

We now explain each of these components in detail.

3.1 Thermal Model of the House

To model the thermal properties of the house, we adopt an approach that is widely used in the home heating literature [4, 10]. In this model, the internal temperature of the home, T_t^{int}, is affected by two antagonistic effects. On the one hand, the heat pump delivers heat at a certain rate that is the product of the electrical power of the pump, r_h, times its thermal efficiency, called *coefficient of performance (COP)*. Mathematically, the heat delivered by the pump is $r_h \cdot COP$, measured in Watt (W). On the other hand, heat leaks from inside the house to the environment at a rate that is proportional to the difference between the internal and external temperatures. The heat loss per time unit depends on the insulation of the house, which is quantified by the leakage rate λ (in W/K). Given this, the instantaneous gain (or loss) of energy in the home at time step t is computed as

$$Q_t = h_t r_h \cdot COP - \lambda \cdot (T_t^{int} - T_t^{ext}) + \epsilon_t, \qquad (1)$$

where ϵ_t is a random variable denoting fluctuations in the heat flow due to random effects not accounted for in the model (e.g., opening doors or windows). Note that Equation (1) is stochastic due to the random effect ϵ_t. However, the thermal properties of the home (i.e. the variables r_h, COP, and λ) can be learned, as demonstrated in [10], assuming that ϵ is independently distributed. Therefore it is sufficient to consider a deterministic version of Equation (1).

[1] Note that all models and techniques presented in this paper can also be applied in a straightforward way to compute an optimal cooling strategy (i.e., to control an air conditioner). However, we restrict ourselves to heating in this paper to simplify the exposition.

The internal temperature at a new time step is then computed as the sum of the previous internal temperature and the heat delivered to (or lost from) the home:

$$T_{t+1}^{int} = T_t^{int} + \frac{Q_t}{c_{air} \cdot m_{air}} \Delta t, \tag{2}$$

where we let c_{air} (unit: J/kg K) and m_{air} (unit: kg) denote the heat capacity and the mass of the air inside the building, respectively.

3.2 The User's Utility Function

Inherent to the home heating problem is the need for the user to trade off *comfort* (i.e., coziness) with the *costs* of heating. Therefore, the optimization has to take both aspects into account. In contrast to most of the prior work in the home heating domain, we follow a decision-theoretic approach and formalize this trade-off with the help of a *utility function*. In this paper, we use the following class of utility functions:

$$u(T_t^{int}, p_t) = \Big(\underbrace{a - b(T_t^{int} - T^*)^2}_{\text{value}} - \underbrace{c(p_t)}_{\text{cost}}\Big)\Delta t, \tag{3}$$

where T^* is the user's most preferred temperature, and p_t the price of electricity. The term $a - b(T^{int} - T^*)^2$ is the value function. The parameter a is the user's willingness to pay for his most preferred temperature (per unit of time), and $b(T^{int} - T^*)^2$ is a quadratic loss function, quantifying the amount of discomfort experienced (per unit of time) due to temperatures deviating from T^*.[2] The parameter b measures the user's sensitivity to temperature deviations.

The cost function $c(p)$ quantifies how much it costs to let the heater run per unit of time. It is given by:

$$c(p_t) = h_t r_h p_t, \tag{4}$$

and is determined by the state of the heater, h_t, the heater's electricity consumption, r_h, and the electricity price, p_t (in Cents/kWh).

4. TEMPERATURE & PRICE PREDICTION

We use GPs to predict future external temperatures as well as electricity prices because GPs are a powerful and flexible framework and have been successfully used to predict external temperatures [10, 6] as well as electricity prices [3]. Our approach is adapted from [10] and [6]. Due to space constraints, we only give a brief overview of GPs. For a more detailed treatment see [9].

4.1 The Prediction Task

Consider a time series $\mathbf{S} = (S(t_1), \ldots S(t_N))$, e.g., for the external temperature. We use the vector notation $\mathbf{t} = (t_1, \ldots, t_N)$ for past time steps, and $\hat{\mathbf{t}} = (\hat{t}_1, \ldots, \hat{t}_T)$ for future time steps for which we want to make predictions. We assume that our training data $\mathbf{y} = (y_{t_1}, \ldots, y_{t_N})$ is distorted by additive i.i.d. Gaussian noise: $y_{t_i} = S(t_i) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$. Given historical data $\mathbf{y}$, we want to make a (probabilistic) prediction of our time series for the next T time steps: $\widehat{\mathbf{S}} = (\widehat{S}(\hat{t}_1), \ldots \widehat{S}(\hat{t}_T))$.

4.2 Gaussian Process Predictions

A Gaussian process approximates the time series $\widehat{\mathbf{S}}$ via a multivariate normal distribution. It is specified by its mean $m(t_i)$ and covariance function $k(t_i, t_j)$. The prior distribution for $\widehat{\mathbf{S}}$ is given by

$$Pr(\widehat{\mathbf{S}}) \sim \mathcal{N}\left(\mathbf{0}, \mathbf{K}(\hat{\mathbf{t}}, \hat{\mathbf{t}})\right), \tag{5}$$

where $\mathbf{K}(\hat{\mathbf{t}}, \hat{\mathbf{t}})$ is the covariance matrix of the input points, i.e. $\mathbf{K}_{i,j} = k(\hat{t}_i, \hat{t}_j)$. The posterior distribution after having learned data points $\mathcal{D} = \{(\mathbf{t}, \mathbf{y})\}$ is computed as

$$\begin{aligned} &Pr(\widehat{\mathbf{S}}|\mathcal{D}) \sim \mathcal{N}(\mathbf{m}_{post}, \mathbf{K}_{post}), \text{ where} \\ &\mathbf{m}_{post} = \mathbf{K}(\hat{\mathbf{t}}, \mathbf{t})\left(\mathbf{K}(\mathbf{t}, \mathbf{t}) + \sigma_n^2 \mathbf{I}\right)^{-1} \mathbf{y}, \text{ and} \\ &\mathbf{K}_{post} = \mathbf{K}(\hat{\mathbf{t}}, \hat{\mathbf{t}}) - \mathbf{K}(\hat{\mathbf{t}}, \mathbf{t})\left(\mathbf{K}(\mathbf{t}, \mathbf{t}) + \sigma_n^2 I\right)\mathbf{K}(\mathbf{t}, \hat{\mathbf{t}}). \end{aligned}$$

[2]Note that we adopted this approach towards modeling discomfort from Rogers et. al [10].

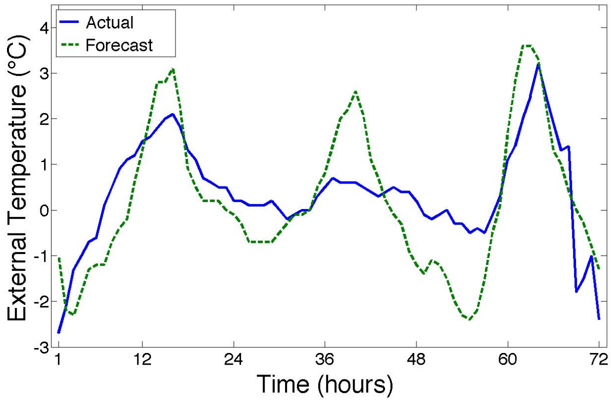


Figure 1: Three days of historical weather data from Zurich.

4.3 External Temperature Prediction

The main idea for the prediction of the external temperature is to train a GP using historical temperature measurements from the actual house as well as weather forecast data from a nearby meteorological service. Obviously, the local weather and the forecast should be highly correlated. In our data, provided by the Swiss national meteorological service MeteoSwiss, the correlation between forecasts and actual temperatures is approximately 0.9. Figure 1 shows a small sample of historical temperature data from Zurich. The green line is the temperature forecast for Zurich from the meteorological service, and the blue line is the actual temperature that was measured in one specific location. As we can see, the two time series are highly (but not perfectly) correlated.

Formally, we consider two temperature time series, one for the local measurements, which we denote as $T^L(t)$, and one for the forecasts, denoted $T^F(t)$. For both, we have historical data (i.e., one data point for every hour), but additionally we have a forecast for the next 24 hours. To use the GPs, we have to specify a model (via the covariance function of the GP) that captures the features of the external temperature sufficiently well. The four features that we model are: (i) daily rise and fall, (ii) rise and fall over longer periods of time (i.e. several days), (iii) erratic movements, and (iv) the correlation between the two time series. The covariance function k_{temp} for two data points (l, t) and (l', t') ($l \in \{Local, Forecast\}$ is the label of the series) is then given by

$$\begin{aligned} k_{temp}\left((l, t), (l', t')\right) = k_1(l, l')\left(k_2(t, t') + k_3(t, t')\right) \\ + k_4(t, t') + k_5(t, t'). \end{aligned} \tag{6}$$

Here, k_1 is a function that measures the cross-correlation between the time series, which is equal to one if the data points are from the same time series, and otherwise equal to θ_1:

$$k_1(l, l') = \begin{cases} 1 & \text{if } l = l', \\ \theta_1 & \text{otherwise.} \end{cases} \tag{7}$$

The covariance function k_2 encodes the daily rise and fall in temperatures. This is modeled using a periodic covariance function with period one day. However, the actual periodicity of weather is only approximately, but not exactly one day. To account for this, we multiply the periodic function with a squared exponential covariance function to allow for more complicated patterns:

$$k_2(t,t') = \theta_2^2 \exp\left(-\frac{(t-t')^2}{2\theta_3^2} - \frac{2\sin^2(\pi(t-t'))}{\theta_4^2}\right). \quad (8)$$

The rise and fall of the temperature over longer periods of time is modeled via a squared exponential covariance function:

$$k_4(t,t') = \theta_5^2 \exp\left(-\frac{(t-t')^2}{2\theta_6^2}\right). \quad (9)$$

The fourth covariance function in Equation (6), k_4, models erratic movements that are uncorrelated between the two time series (e.g. due to specific conditions at the measurement site). For this, we use a Matern class kernel as it is able reproduce such fluctuating temperature movements:

$$k_4(t,t') = \delta_{l,l'}\theta_7^2\left(1+\frac{\sqrt{3}(t-t')}{\theta_8}\right)\exp\left(-\frac{\sqrt{3}(t-t')}{\theta_9}\right). \quad (10)$$

To account for measurement noise, which we assume to be i.i.d additive Gaussian, we use the following noise covariance function:

$$k_5(t,t') = \theta_{10}^2\delta_{t,t'}, \quad (11)$$

where $\delta_{t,t'}$ is the Kronecker delta between time points. Note that $\theta_1,\ldots,\theta_{10}$ are hyper-parameters of the GP whose values must be determined using maximum likelihood estimation.

4.4 Electricity Price Prediction

We model our price function according to characteristics found in spot market prices. According to Weron [17], a salient feature of daily prices is the periodicity: an increase in the morning (when people wake up), a decrease in the afternoon, and another increase in the evening (when people return home). We model this using a periodic covariance function with period half a day. As before, we allow for deviations from exact periodicity by multiplying the periodic covariance function with a squared exponential:

$$k_6(t,t') = \theta_{11}^2 \exp\left(-\frac{(t-t')^2}{2\theta_{12}^2} - \frac{2\sin^2(\pi(t-t'))}{\theta_{13}^2}\right). \quad (12)$$

The second feature of price movements are the erratic price fluctuations, which we model by a Matern class kernel:

$$k_7(t,t') = \sigma_6^2\left(1+\frac{\sqrt{3}(t-t')}{\theta_{14}}\right)\exp\left(-\frac{\sqrt{3}(t-t')}{\theta_{15}}\right). \quad (13)$$

The covariance function for the price is the sum of k_3 and k_4 plus a noise term $k_8(t,t') = \theta_{16}^2\delta_{t,t'}$:

$$k_{price}(t,t') = k_6(t,t') + k_7(t,t') + k_8(t,t'). \quad (14)$$

Again, $\theta_{11},\ldots,\theta_{15}$ are hyper-parameters of the GP whose values must be determined using maximum likelihood estimation.

5. HOME HEATING MDP

We now formalize the home heating problem as an MDP. An MDP is defined by a tuple (S, A, T, R), where S is the state space, A is the action space, $T : S \times S \times A \to \mathbb{R}_+$ is the state transition function, and $R : S \times A \to \mathbb{R}$ the reward function. We consider an MDP with a finite horizon of one day.[3] Defining the states, actions and the reward function is quite straightforward. The difficulty lies in computing the state transition probabilities, which is where the information obtained from the GPs is used.

States: The state space consists of the Cartesian product $S = \mathcal{T}^{int} \times \mathcal{T}^{ext} \times \mathcal{P} \times TIME$, where $\mathcal{T}^{int}$ and $\mathcal{T}^{ext}$ are the sets of internal and external temperatures, respectively, $\mathcal{P}$ the set of prices, and *TIME* the set of time steps for one day. Both, the prices and the temperatures are discretized, which is quite natural for the prices (in Cents), and also for the temperatures, since humans cannot notice the difference between two temperatures given a small enough level of granularity (e.g., between 22.0 and 22.5 degrees Celsius). To simplify the exposition, we denote the state $s = (T^{int}, T^{ext}, p, t)$ as $s_t = (T_t^{int}, T_t^{ext}, p_t)$.

Actions: The action space is $A = \{0, 1/(N_A-1), 2/(N_A-1), \ldots, 1\}$, where N_A is the number of actions available. For example, if $N_A = 2$, then $A = \{0, 1\}$, which corresponds to the heater being off or on, respectively, i.e., setting $h_t = 0$ or $h_t = 1$.

Reward Function: The reward function is simply the user's utility function, i.e., the user's value for a certain internal temperature T_t^{int} minus the cost of heating:

$$R(s_t, h_t) = \left(a - b(T^* - T_t^{int})^2 - h_t r_h p_t\right) \cdot \Delta t. \quad (15)$$

State Transition Function: The state transition function is a function that specifies, for every triple $(s_t, s_{t+1}, a) \in S \times S \times A$, the probability of arriving at state s_{t+1} if action h_t is taken in state s_t:

$$T(s_{t+1}, s_t, h_t) = Pr\left((T_{t+1}^{int}, T_{t+1}^{ext}, p_{t+1})|(T_t^{int}, T_t^{ext}, p_t), h_t\right). \quad (16)$$

Although computing these probabilities might seem daunting at first, we can greatly simplify this task by making a few observations. First, note that T_{t+1}^{ext} and p_{t+1} are independent of T_t^{int} and h_t; and also of p_t and T_t^{ext}, respectively.[4] Furthermore, T_{t+1}^{int} does not depend on p_t. Exploiting these independencies, we can write:

$$T(s_{t+1}, s_t, h_t) = Pr\left(T_{t+1}^{int}|T_t^{int}, T_t^{ext}, h_t\right) Pr\left(T_{t+1}^{ext}|T_t^{ext}\right) Pr\left(p_{t+1}|p_t\right). \quad (17)$$

Note that the state transition for the internal temperature is deterministic by assumption and can be derived via Equation (2). However, the evolution of the external temperature and the price is stochastic. Therefore, we derive the transition probabilities $Pr(T_{t+1}^{ext}|T_t^{ext})$ and $Pr(p_{t+1}|p_t)$ using the probabilistic information gathered from the GPs.

5.1 Transition Probabilities for External Temperatures and Prices

We now describe how to derive the transition probabilities for the external temperatures. The approach is completely analogous for the electricity prices.

Recall that the GP gives us a predictive distribution for $\widehat{\mathbf{T}}^{ext} = (\widehat{T}^{ext}(\hat{t}_1), \ldots, \widehat{T}^{ext}(\hat{t}_{|TIME|}))$ that is a multivariate normal distribution

$$Pr\left(\widehat{\mathbf{T}}^{ext}|\mathcal{D}\right) \sim \mathcal{N}(\mathbf{m}, \mathbf{K}). \quad (18)$$

[3] We use a finite-horizon MDP for two reasons. First, the thermal effects of heating and heat leakage manifest themselves within minutes to hours. Therefore, optimizing the heating now will not greatly affect the heating for tomorrow or even further away. Second, an infinite horizon MDP would assume a stationary model of the external temperature and electricity prices. However, it is much better to predict the external temperature and electricity prices using day-ahead forecasts, thus dropping the stationarity assumption.

[4] Of course, the price may in practice also depend on the external temperature because weather conditions influence demand for energy and therefore, if many people have to heat a lot at the same time, prices may increase. However, we ignore this dependency to simplify the exposition.

Algorithm 1: Home Heating Algorithm

Input: utility function u
Variables: internal temperature T_t^{int}, external temperature T_t^{ext}, price p_t, optimal heating policy π^*
begin
  foreach *day* **do**
    $\widehat{P} \leftarrow GP.predictPrices()$
    $\widehat{T} \leftarrow GP.predictExternalTemperature()$
    $M \leftarrow$ **new** $MDP(u, \widehat{P}, \widehat{T})$
    $M.computeTransitionProbabilities()$
    $\pi^* \leftarrow M.computeOptimalPolicy()$
    for *t=1 to # of time steps per day* **do**
      $(T_t^{int}, T_t^{ext}, p_t) \leftarrow M.updateEnvironment()$
      $M.heatOptimally(\pi^*, T_t^{int}, T_t^{ext}, p_t)$

For the state transition function we need to compute conditional probability distributions of the form

$$Pr\left(\widehat{T}^{ext}(\hat{t}_i) = T | \widehat{T}^{ext}(\hat{t}_{i-1}) = \widetilde{T}, \mathcal{D}\right) \quad (19)$$

for all $i = 2, \ldots, |TIME|$ and $T, \widetilde{T} \in \mathcal{T}^{ext}$. We perform these computations in two steps: First, we compute the conditional distribution of $\widehat{T}^{ext}(\hat{t}_i)$ given $\widehat{T}^{ext}(\hat{t}_{i-1})$. Second, we integrate the conditional distribution to obtain a discrete conditional probability distribution.

Step 1: The conditional distribution can be computed as follows:

$$\begin{aligned} &Pr\left(\widehat{T}^{ext}(\hat{t}_i) | \widehat{T}^{ext}(\hat{t}_{i-1}) = \widetilde{T}, \mathcal{D}\right) \sim \mathcal{N}(m_{cond}, \sigma_{cond}), \text{ with} \quad (20)\\ &m_{cond} = m_i + \frac{\mathbf{K}_{i,i-1}}{\mathbf{K}_{i-1,i-1}} \cdot (\widetilde{T} - m_{i-1}), \text{ and} \\ &\sigma_{cond} = \mathbf{K}_{i,i} - \frac{(\mathbf{K}_{i,i-1})^2}{\mathbf{K}_{i-1,i-1}}. \end{aligned}$$

Step 2: We then integrate the conditional distribution over the interval $[T - \alpha, T + \alpha]$, where α is half of the discretization size in the temperature space, to obtain our final discretized transition function for the external temperature:

$$Pr\left(T^{ext}_{t_i^*} = T | T^{ext}_{t_{i-1}^*} = \widetilde{T}\right) = \int_{T-\alpha}^{T+\alpha} Pr\left(T^{ext}_{t_i^*} = y | T^{ext}_{t_{i-1}^*} = \widetilde{T}, \mathcal{D}\right) dy. \quad (21)$$

For example, if the discretization is $\mathcal{T}^{ext} = \{0, 1, 2, \ldots\}$ and we would like to compute the probability that the external temperature is 6°C after being 5°C, then

$$Pr\left(\widehat{T}^{ext}(\hat{t}_i) = 6 | \widehat{T}^{ext}(\hat{t}_{i-1}) = 5\right) = \int_{5.5}^{6.5} Pr\left(\widehat{T}^{ext}(\hat{t}_i) = y | \widehat{T}^{ext}(\hat{t}_{i-1}) = 5, \mathcal{D}\right) dy.$$

Finally, we normalize all probabilities computed this way to obtain a correct conditional probability distribution.

5.2 Computing an Optimal Policy

Now that we have constructed all components of the MDP, we can compute an optimal policy using dynamic programming (DP). Note, that at every iteration of the DP algorithm, one must take care to only include the reachable states in the time dimension (i.e., only consider states that are one time step earlier). The resulting optimal policy π^* corresponds to the optimal value function V^* that solves the Bellman optimality equation:

$$V^*(s) = \max_a \left(R(s, a) + \sum_{s'} Pr(s'|s, a) V^*(s') \right). \quad (22)$$

Thus, the optimal policy prescribes the action that maximizes the sum of the one-step reward and the expected utility going forward, assuming that the optimal policy is followed in the future. A summary of the whole heating algorithm is provided in Algorithm 1.

6. EXPERIMENTS

We evaluate our MDP-based heating algorithm via two simulation experiments. In Experiment I, we consider a simple heater that is either switched on or off. In Experiment II, we consider a heater that can work at any level between zero and maximum power.

We consider two different pricing scenarios: *times-of-use pricing* and *real-time pricing*. Times-of-use pricing models a situation in which the electricity provider sets fixed prices for certain specified (and fixed) periods of the day. Under real-time pricing the electricity price changes according to real-time market conditions. Because the actual demand and supply of energy depends on many factors (e.g., available utilities and weather conditions), real-time prices can only be predicted with a high level of uncertainty.

6.1 Experiment I: MDP vs. MIP

For Experiment I, we consider a heat pump that can only be switched on or off. We compare our MDP-based algorithm against three benchmark algorithms: a conventional thermostat that implements a simple rule-based heating policy, and a mixed integer program (MIP) that comes in two versions: one that aims to minimize heating costs, and another one that maximizes the user's utility.

Conventional Thermostat. A conventional thermostat tries to keep the room temperature around a set temperature T_{set} by implementing the following rule:

$$h_t^{thermostat} = \begin{cases} 0 & \text{if } T_{t-1}^{int} > T^{set} + \Delta T \\ 1 & \text{if } T_{t-1}^{int} < T^{set} - \Delta T \\ h_{t-1}^{thermostat} & \text{otherwise} \end{cases} \quad (23)$$

Mixed-Integer Program. Our second benchmark algorithm is a MIP, introduced by Rogers et. al [10].[5] The MIP minimizes the heating costs, subject to the constraint that the cumulative discomfort does not exceed a maximum discomfort level. Discomfort is measured as a quadratic loss function as in the utility function defined in Equation (3). We let $h_t \in \{0, 1\}$ denote the decision variables, c_t the cost of heating, d_t the discomfort, and D_{max} the maximum discomfort level.[6] The whole MIP can be stated as:

$$\begin{aligned} \min \ & \sum_t h_t c_t \quad (24) \\ \text{s.t.} \ & Q_t = h_t r_h COP - \lambda \cdot (T_t^{int} - T_t^{ext}) \\ & T_{t+1}^{int} = T_t^{int} + \frac{Q_t}{c_{air} \cdot m_{air}} \Delta t \\ & d_t = (T^* - T_t^{int})^2 \\ & \sum_t d_t \le D_{max} \end{aligned}$$

Note that this approach implicitly assumes that prices and external temperatures are known in advance. In particular, only the mean predictions from the GP are used instead of the whole distribution.

We also consider a modified version of this MIP, where the objective function is changed to now maximize the user's utility, which is defined analogously to the reward function of the MDP (see Equation (15)). Additionally, we drop the constraint that the discomfort should not fall below a certain target discomfort level.

[5] We thank the authors of [10] for providing their CPLEX code.
[6] D_{max} is set to the level of discomfort the user would experience if a conventional thermostat were run instead of the MIP.

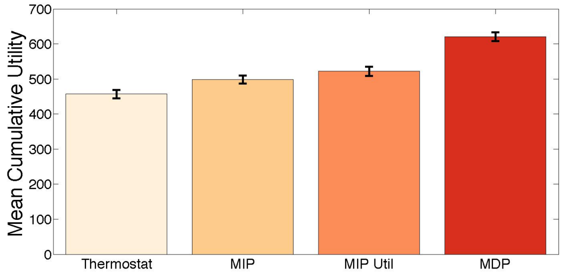


Figure 2: Results for real-time pricing. The bars denote (from left to right): the thermostat, the MIP that minimizes cost, the MIP that maximizes utility, and the MDP. Performance is measured as the mean cumulative utility, averaged over the month of January 2013.

6.1.1 Experimental Set-up

We let every algorithm heat sequentially for 31 days (i.e., for one month). Each day consists of 144 time steps, which corresponds to 10 minute time intervals. We use weather data for Zurich from December 2012 to January 2013, provided by the Swiss national meteorological service MeteoSwiss, which contains hourly forecasts for Zurich as well as actual measurements for one specific location in Zurich. We use the data from December to train the GPs, and the one from January for the actual experiment.[7]

In the times-of-use pricing scenario, there are three tariffs: 10, 20, and 40 cents/kWh. The prices are known a priori to all algorithms. In the real-time pricing scenario, the prices are generated using a GP that has the same model as described in section 4.4 to produce synthetic, but realistic pricing data. The GPs used for prediction are trained on the (synthetic) price data from December.

For the user's utility function, we set the parameters $a = 8/\Delta t$ and $b = 1/\Delta t$. These values are chosen such that values and costs in the reward function of the MDP (see Equation (15)) are approximately of the same size, corresponding to a user for whom, at typical prices, comfort and costs have comparable magnitudes.

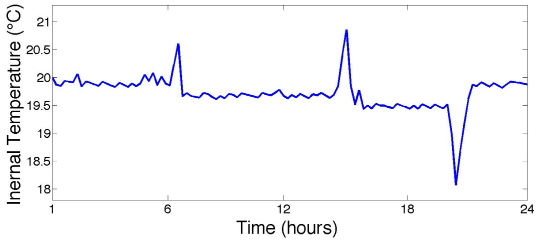
The dimensions of the house are $1,000\,m^3$, the mass of air in the house $m_{air} = 1,205\,kg$, the leakage rate $\lambda = 90\,W/kg$, the heat capacity $c_{air} = 1,000\,J/kg/K$. The power of the heater is $r_h = 1,500\,W$ with a $COP = 2.5$. The values are adopted from [10] and correspond to a small, well insulated home.

For the MDP, we discretize the temperature in steps of $0.5°C$ and the prices in steps of 5 cents. This set-up results in an MDP with approximately 1 million states that can be solved optimally in a few seconds on a standard PC. We use IBM ILOG CPLEX to solve the MIP and adopt the same approach as in [10] restricting CPLEX to run for 5 minutes per problem instance. The solver produces iteratively improving solutions. Thus, if the MIP does not terminate within the 5 minutes, it returns the best solution found so far.

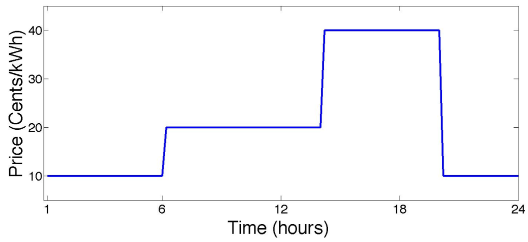
6.1.2 Results and Discussion

Figure 2 shows the results for the real-time pricing scenario (the results for the times-of-use scenario are qualitatively the same). We report the average cumulative utility (and standard errors) achieved by the different algorithms. We see that the MDP-based algorithm provides significantly higher average utility compared to all other approaches, improving the utility by more than 15%.

[7]The root mean squared errors of the GP and the meteorological service are $RMSE_{GP} = 1.29$ and $RMSE_{MF} = 1.61$, respectively. The approximately 20% improvement in accuracy makes sense since the GP can adapt to the peculiar climatic conditions (e.g., trees that provide shade) at the specific location whereas the forecast does not include this information.



(a)



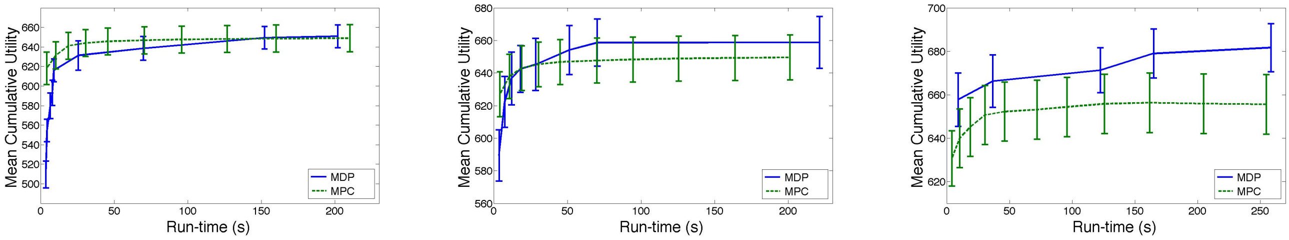
(b)

Figure 3: (a) Temperature profile over the course of one day, when running the MDP. (b) The corresponding times-of-use tariff.

There are several observations to discuss. First, we see that the conventional thermostat performs worst. This makes sense, because it completely neglects the cost component of the user's utility function. Second, the cost-minimizing MIP performs slightly better than the thermostat. Recall that it computes cost-minimizing heating plans that do not to exceed a certain level of discomfort. Third, we see that the utility-maximizing MIP performs better than the cost-minimizing MIP, which demonstrates that cost-minimization subject to comfort constraints only imperfectly approximates the maximization of the user's utility. However, even the utility-based MIP is still significantly worse than the MDP. This is because the MIP implicitly assumes that the predictions for the external temperatures and for the prices are perfectly accurate. If this assumption is not correct, then the MIP leads to sub-optimal decisions, e.g., not pre-heating when prices are low, or not saving energy when the outside temperature is about to rise, which leads to significantly higher discomfort or costs, compared to our MDP-based algorithm.

To illustrate the MDP-based approach, Figure 3(a) provides an example of the internal temperature profile that results from executing the MDP-based heating policy, while the corresponding times-of-use prices are shown in Figure 3(b).[8] By tracing the temperature curve over the course of the day, we gain insights into how the MDP optimizes the trade-off between comfort and costs. We see that just before the price goes up at the 6-hour mark, the MDP preheats a little bit, exploiting the low prices. It then uses less energy than before, consequently leading to a slightly lower temperature. Just before the next price increase at the 14-hour mark, it pre-heats again, exploiting the 20 cents/KWh price. Over the next six hours it uses even less energy than in the previous eight hours, leading to an even lower temperature. Just before the price goes back to 10 cents/KWh, the MDP essentially stops heating (to conserve costs in the high price regime), which leads to a momentary drop in temperature. Once the low price regime is reached, normal heating resumes, and the temperature goes back to the original level.

[8]Note that to produce the graph in Figure 3(a), we considered the scenario from Experiment II where the heater can be set to different levels between zero and maximum power. In particular, we used an MDP with 25 actions instead of just *on/off*, because the resulting temperature curve more cleanly illustrates the MDP policy.



(a) $b = 2$ (b) $b = 1$ (c) $b = 0.1$

Figure 4: A comparison between MDPs and MPCs. The graphs show the mean cumulative utilities for the MDP (solid blue line) and the MPC (dotted green line), for different values of the sensitivity parameter b (a smaller b corresponds to a more cost-sensitive user).

6.2 Experiment II: MDP vs.MPC

For Experiment II, we assume that the heater can work at any level between zero and maximum power. We compare our MDP-based algorithm (now with more than two actions) to an approach that uses model predictive control [18], which has proven successful in the heating domain [4, 5].

Model Predictive Control. MPCs are online algorithms that iteratively solve an optimization problem for a given time horizon to find the best sequence of (continuous) control actions, but only apply the first action to the system. After each time step, the system state is observed and a new optimization problem is solved given the new state. Thus, the time horizon is shifted one time step into the future. Applied to the home heating problem, this means that at every time step, we solve the following optimization problem:

$$\max_{h_t} \sum_t \Big(a - b(T^* - T_t^{int})^2 - h_t r_h p_t\Big) \cdot \Delta t, \text{ s.t.} \qquad (25)$$

$$Q_t = h_t r_h COP - \lambda \cdot (T_t^{int} - T_t^{ext})$$

$$T_{t+1}^{int} = T_t^{int} + \frac{Q_t}{c_{air} \cdot m_{air}} \Delta t$$

$$h_t \in [0, 1].$$

This optimization problem is a quadratic program, which can be solved very efficiently. As in Experiment I, the time horizon is set to 24 hours. The predictions for the external temperature and prices are computed via GPs, in the same manner as was done for the MIPs. However, in contrast to the MIP-based approach, the GPs are updated using the new measurements of the external temperature and the current price made available at the end of each time step.

This particular version of an MPC is called *certainty equivalent MPC (CE MPC)*, because the external temperature and the electricity prices are set to the values predicted by the GPs, ignoring probabilistic effects. However, this loss of information is countered by the fact that the MPC works in an online fashion.

Note that MPCs work similarly to MDPs in the sense that MDPs *solve* the Bellman optimality equation (see Equation (22)) for a discretized version of the problem, and MPCs *approximate* the Bellman optimality equation for a continuous version of the problem. However, it is also informative to consider in more detail how the two methods differ. First, while the MDP computes an optimal policy that provides the optimal action for *every* state, the CE MPC only yields a policy for the states it believes it will encounter, given its model, the initial conditions, and the current predictions. Second, the bulk of the MDP computations are performed offline (i.e., at the beginning of the day), while the computations for the MPC (i.e. updating predictions, solving the optimization problem) must be repeated every time step. Third, the run-time of the MDP (given a fixed problem size) grows polynomially with the number of time steps, actions, and states. Thus, if we increase the discretization granularity for all three components simultaneously, then the run-time grows cubically. For the MPC, the trade-off between run-time and performance is less pronounced since only the time discretization matters. Finally, MDPs offer a rich language to model sequential decision making under uncertainty, whereas MPCs only have limited ability to model probabilistic environments.[9]

6.2.1 *Experimental Setup*

For both, the MDP and the MPC, we need to make a trade-off between computational complexity and performance: the finer the discretizations (time, actions, and states for the MPD; time for the MPC), the better the performance, but also the higher the computational burden. In this section, we study this trade-off in detail.

The basic experimental setup is similar to the one used in the real-time pricing scenario of Experiment I. We run every algorithm for 18 days and report average cumulative utility. However, we use different discretizations for the MDP and the MPC corresponding to different run-times. For the MPC, we vary the number of time steps from 24 to 192. For the MDP, we increase the number of time steps from 24 to 192, and the number of actions from 2 to 20, and report the highest utility achieved for a particular run-time.

We also vary the sensitivity parameter b of the utility function (see Equation (15)). We consider three values for b that corresponds to three different types of users: a value of $b = 2$ corresponds to a comfort-sensitive user; a value of $b = 1$ corresponds to an approximately equal weighting of comfort and cost; a value of $b = 0.1$ corresponds to a cost-sensitive user.

6.2.2 *Results and Discussion*

Figure 4 shows the results of Experiment II. The solid blue line and the dotted green line correspond to the MDP and the MPC, respectively. The graphs plot the average cumulative utility per day (on the y-axis) versus the average time spent to compute the optimal heating policy for one day (on the x-axis).

First, we see that for $b = 2$ and $b = 1$, there is no statistically significant difference between the expected utility achieved by the MDP and the MPC, except at very low run-times (less than 10 seconds), where the MPC outperforms the MDP. For cost-sensitive users ($b = 0.1$), the MDP leads to higher expected utility than the MPC, and this difference is statistically significant for run-times larger than 150 seconds. This results makes sense: we expect the

[9] There also exist *stochastic MPCs* that can handle some forms of uncertainty. At the same time, there also exist more sophisticated methods for handling continuous state and/or action spaces for MDPs that avoid some of the limitations of a discretized state or action space. Furthermore, one could also consider online (roll-out) methods for the MDP, which could speed up the computations and which would also allow the MDP to update the GP predictions based on the new information in each time step. All of these considerations are left to future research.

MDP to be better at saving costs because its probabilistic model enables it to better account for the stochastic prices and temperatures.

Overall, we see that the performance of both algorithms improves a lot in the beginning as the computational complexity is increased, but that the rate of improvement quickly diminishes. This effect is particularly strong for the MDP, which makes sense, because the MDP is severely limited at very low run-times (with a low level of discretization in three dimensions).

7. CONCLUSION AND FUTURE WORK

In this paper, we have studied the adaptive home heating problem under weather and price uncertainty. We have presented a general technique that uses the predictive distributions obtained from GP regressions to construct the state transition probabilities of a corresponding MDP. Applied to the heating domain, the solution to the resulting home heating MDP constitutes a sequentially optimal heating policy that accounts for all available probabilistic information. Via simulations, we have demonstrated that in a scenario where the heater is limited to being switched on or off, our approach outperforms all benchmark algorithms from the literature. For another scenario, where a heater can run at any level between zero and maximum power, we have compared our MDP-based approach against an MPC-based approach. In particular, we have studied the resulting trade-off between computational run-time and performance for MDPs and MPCs. Our results indicate that both algorithms lead to very similar performance, except at very low run-times, where the MPC is slightly better. However, for price-sensitive users, the MDP eventually leads to significantly higher expected utility than the MPC, because it is better to able to account for the stochastic nature of energy prices and outside temperatures.

One important advantage of our MDP-based solution is that it naturally lends itself towards incorporating our prior work on preference elicitation in the home heating domain [13]. For this paper, we have assumed that we already have a good model of the user's utility function. In practice, however, this model must be learned over time, while the thermostat already optimizes the heating policy. Towards this end, our future work will involve extending our MDP model to explicitly incorporate the preference elicitation decisions. Because the user's utility is never fully revealed to the thermostat, this leads to a partially-observable MDP (i.e., POMDP) [1]. In our future research, we will work towards this vision of a smart thermostat that acts optimally on the user's behalf by carefully eliciting the user's preferences while simultaneously computing an optimal heating policy that maximizes the user's expected utility.

Acknowledgments

We would like to thank Siddhartha Ghosh for insightful discussions and the anonymous reviewers for helpful comments. This research was supported by the Swiss National Science Foundation (SNSF).

8. REFERENCES

[1] C. Boutilier. A POMDP Formulation of Preference Elicitation Problems. In *Proceedings of National Conference on Artificial Intelligence (AAAI)*, 2002.

[2] P. Cramton and A. Ockenfels. Economics and Design of Capacity Markets for the Power Sector. Working Paper, University of Maryland, May 2011.

[3] D. J. Leith, M. Heidl, and J. V. Ringwood. Gaussian Process Prior Models for Electrical Load Forecasting. In *IEEE International Conference on Probabilistic Methods Applied to Power Systems*, 2004.

[4] Z. Y. L. McLaughlin, L. Jia, M. C. Murphy-Hoye, A. Pratt, and L. Tong. Modeling and Stochastic Control for Home Energy Management. In *Proceedings of the Power and Energy Society (PES) General Meeting*, 2012.

[5] F. Oldewurtel, A. Parisio, C. N. Jones, M. Morari, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, and K. Wirth. Energy Efficient Building Climate Control Using Stochastic Model Predictive Control and Weather Predictions. In *American Control Conference (ACC), 2010*, pages 5100–5105. IEEE, 2010.

[6] M. A. Osborne, S. Roberts, A. Rogers, S. Ramchurn, and N. R. Jennings. Towards Real-time Information Processing of Sensor Network Data Using Computationally Efficient Multi-output Gaussian Processes. In *Proceedings of the 7th international conference on Information processing in sensor networks*, pages 109–120. IEEE Computer Society, 2008.

[7] J. Palmer and I. Cooper. Great Britain's Housing Energy Fact File. Departmant of Energy and Climate Change (DECC), 2011.

[8] S. Ramchurn, P. Vytelingum, A. Rogers, and N. R. Jennings. Putting the "Smarts" into the Smart Grid: A Grand Challenge for Artificial Intelligence. *Communications of the ACM*, 55(4):86–97, 2012.

[9] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2005.

[10] A. Rogers, S. Maleki, S. Ghosh, and J. Nicholas R. Adaptive Home Heating Control Through Gaussian Process Prediction and Mathematical Programming. In *Proceedings of the 2nd International Workshop on Agent Technology for Energy Systems (ATES)*, 2011.

[11] A. Rogers, S. D. Ramchurn, and N. R. Jennings. Delivering the Smart Grid: Challenges for Autonomous Agents and Multi-agent Systems Research. In *Proceedings of Conference on Artificial Intelligence (AAAI)*, 2012.

[12] J. Scott, A. Bernheim Brush, J. Krumm, B. Meyers, M. Hazas, S. Hodges, and N. Villar. PreHeat: Controlling Home Heating Using Occupancy Prediction. In *Proceedings of the 13th International Conference on Ubiquitous Computing (UbiComp)*, 2011.

[13] M. Shann and S. Seuken. An Active Learning Approach to Home Heating in the Smart Grid. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*. 2013.

[14] U.N. Framework Convention on Climate Change. Kyoto Protocol. http://unfccc.int/kyoto_protocol/items/2830.php, December 1997.

[15] U.S. Energy Information Administration. Energy Consumption in Homes by End Users. http://www.eia.gov/consumption/residential, March 2013.

[16] P. Vytelingum, T. D. Voice, S. D. Ramchurn, A. Rogers, and N. R. Jennings. Agent-based Micro-storage Management for the Smart Grid. In *Proceedings of International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2010.

[17] R. Weron. *Modeling and Forecasting Electricity Loads and Prices: A Statistical Approach*. HSC Books. Hugo Steinhaus Center, Wroclaw University of Technology, 2006.

[18] J.B. Rawlings and D.Q. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill Publishing, Madison WI, 2009.

4 Save Money or Feel Cozy? A Field Experiment Evaluation of a Smart Thermostat

The content of this chapter has previously appeared in:

> Shann, M., Alan, A., Seuken, S., Costanza, E., and Ramchurn, S. D. (2017). Save Money or Feel Cozy? A Field Experiment Evaluation of a Smart Thermostat that Learns Heating Preferences. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '17, São Paulo, Brazil.

Save Money or Feel Cozy? A Field Experiment Evaluation of a Smart Thermostat that Learns Heating Preferences

Mike Shann[1], Alper Alan[2], Sven Seuken[1], Enrico Costanza[3], Sarvapali D. Ramchurn[2]
[1]University of Zurich, Zurich, Switzerland, {shannm, seuken}@ifi.uzh.ch
[2]University of Southampton, Southampton, United Kingdom, {a.t.alan, sdr1}@soton.ac.uk
[3] University College London, London, United Kingdom, {e.costanza@ucl.ac.uk

ABSTRACT

We present the design of a fully autonomous smart thermostat that supports end-users in managing their heating preferences in a real-time pricing regime. The thermostat uses a machine learning algorithm to learn how a user wants to trade off *comfort* versus *cost*. We evaluate the thermostat in a field experiment in the UK involving 30 users over a period of 30 days. We make two main contributions. First, we study whether our smart thermostat enables end-users to handle real-time prices, and in particular, whether machine learning can help them. We find that the users trust the system and that they can successfully express their preferences; overall, the smart thermostat enables the users to manage their heating given real-time prices. Moreover, our machine learning-based thermostats outperform a baseline without machine learning in terms of usability. Second, we present a quantitative analysis of the users' economic behavior, including their reaction to price changes, their price sensitivity, and their comfort-cost trade-offs. We find a wide variety regarding the users' willingness to make trade-offs. But in aggregate, the users' settings enabled a large amount of demand response, reducing the average energy consumption during peak hours by 38%.

Keywords

Sustainability; home heating; real-time prices; user interfaces; machine learning; field experiment.

1. INTRODUCTION

Over the last decade, we have witnessed a steadily increasing effort to realize a paradigm shift in the energy sector. The goal of this shift is to transform energy production from a centralized architecture of power plants that burn the ever dwindling amounts of fossil fuels to a distributed grid of renewable energy sources like wind and solar [30]. This transformation of the electricity grid is motivated by the need to combat the negative economic and sociological effects of climate change as well as by the fact that the production of many conventional oil and gas fields are decreasing.

Implementing such a distributed electricity grid poses a number of challenges due to the current structure of the grid and the volatility in the production of renewable energy. If the share of renewable energy sources keeps growing, maintaining the stability of the grid will become an increasingly challenging problem since the production level of renewables is very hard to control and therefore, matching supply and demand will become much more difficult [11].

Appears in: *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*

1.1 Managing the Demand for Energy

To solve the problem of grid stability, it will be critical to also *manage the demand side* by incentivizing consumers to adapt their consumption levels to the amount of energy available in the grid [20]. One way to encourage consumers to decrease their demand when energy is scarce is via financial incentives [3]. A particular financial mechanism that has been put forward is *real-time pricing*, where the price of electricity varies across the day according to market forces. Real-time pricing has a number of advantages over flat pricing. First, economists argue that real-time pricing improves system reliability and mitigates market power in the long term [6]. Second, it offers consumers the opportunity to save significant amounts of money if they are willing to dynamically adjust their consumption [12]. A number of power companies in the US and Europe have successfully conducted pilot studies, to assess the potential benefits and the feasibility of using real-time pricing for *residential end-users* (see e.g., [13, 14, 15, 29]). Some power companies already offer real-time pricing programs to their end-users.[1]

While energy plays a large role in many domains, residential heating is one of the major drivers of energy consumption, accounting for approximately 45% and 62% of the total household energy consumption in the US and the UK, respectively, which amounts to 10% and 18% of the respective country's total energy consumption [19, 31]. With the goal in mind to move away from fossil fuels, the electrification of heating using heat pumps is seen as a key technology for achieving a society that is more sustainable. Indeed, many low-carbon scenarios assume that in the future, a majority of houses will be heated by heat pumps (see, e.g., [9]). These reasons make home heating a formidable case study to explore the potential for demand-side management with real-time electricity prices.

1.2 Home Heating with Smart Thermostats

In this paper, we envision a future electricity grid where a substantial number of private homes are heated by heat pumps and at least some end-users are exposed to real-time prices. Obviously, this poses multiple challenges for the design of a usable heating system.

First, it is not feasible for end-users to constantly monitor the energy price and manually adjust their thermostat whenever prices change. Thus, we need an autonomous agent, which we call the *smart thermostat*, that *automatically* reacts to price changes on the user's behalf. Second, before the smart thermostat can make these decisions autonomously, it needs to know how the user wants to trade off *comfort* (heating to a particular temperature) versus *cost* (for heating to that temperature) at different price levels. Some users might be willing to spend a lot of money to have their home always heated to a comfortable temperature, while others may want

[1]E.g., Commenwealth Edison's "Residential Real-time Pricing Program": `https://rrtp.comed.com/`

to decrease their temperature if energy becomes too expensive. This means that to achieve high *economic efficiency*, it must be possible to personalize the smart thermostat to individual users. However, manually specifying how to trade off comfort and cost at all price levels might lead to high *cognitive costs* on the user's side.

Obviously, there is a tension between economic efficiency on the one hand high cognitive costs on the other hand. To address this tension, we turn to the *hidden market design* paradigm introduced by Seuken et al. [24], who argued that it is often necessary to *hide* some of the market's complexity from the end-users. They showed that a hidden market UI can reduce the interaction complexity for the end-users, while still maintaining the loop between the market and the users [25]. In our domain, we instantiate the hidden market design paradigm by designing a smart thermostat that elicits the user's trade-off between comfort and cost over time while keeping the user's input at a minimum. To realize this, we build on prior work by Shann and Seuken [26] who proposed a *machine learning algorithm* to solve this exact problem. However, their work was purely theoretical. In particular, they did not design any UIs or a real system. In this research project, we expand on this theoretical work by designing a real-world application of a smart thermostat that supports users in managing their heating preferences in a real-time pricing regime. We deployed this smart thermostat in 30 homes in the UK and ran a 30-day field experiment from February to March 2015 to explore how people interact with such a system.

1.3 Overview of Contributions

We make two main contributions. First, we study whether our smart thermostats can *enable* end-users to successfully handle real-time prices in the home heating domain – in particular, whether using machine learning can *improve the usability* of the thermostat. Our results show that the majority of our users were satisfied with the smart thermostats, and trusted them to automatically adjust the temperature for them. More importantly, the data shows that the machine learning algorithm increased the usability of the system, compared to a baseline implementation that uses no learning.

Second, we present a detailed quantitative analysis of the *economic behavior* of our 30 participants when exposed to real-time pricing. Our results show that the users react to price changes in an economically rational way, and on average, they are willing to decrease their indoor temperature by 3 °C when energy is most expensive. Fortunately, due to the thermal inertia of the homes, the indoor temperature does not decrease by more than 1 °C, even during peak price hours. Still, this price-sensitive behavior leads to a large amount of demand response, reducing the average energy consumption by 38% during peak hours.

2. RELATED WORK

Automated Control in the Smart Grid. Yang et al. [32] examined the real-world uptake of a smart thermostat with 23 participants. They highlighted how sub-optimal decisions taken by a smart thermostat are likely to cause user frustrations and may lead them to abandon the technology. Bourgeois et al. [8] deployed energy-aware washing machines in 18 households and found that sending suggestions on when to do the laundry via text messages is more effective than other interventions. Costanza et al. [10] conducted a field experiment with 10 participants that used "Agent B," an agent that helps users book their washing machine given real-time prices. Their results indicate that users are willing to shift their washing in response to real-time prices. Alan et al. [1] tested "Tariff Agent," an agent that helps users select electricity tariffs on a daily basis, in a field experiment with 10 users. The results show that people are willing to delegate decisions regarding energy consumption to an agent.

Our study differs from the above studies in two key ways. First, our system is *fully autonomous*, i.e., it takes decisions on the users' behalf instead of just giving advice to the users. Second, the system's decisions have a direct impact on users' well-being via the temperature it sets in the respective homes, while previous systems only affected the study participants' financial rewards.

In our own prior work [2], we already analyzed the exit interviews with the 30 participants of our field experiment from an HCI perspective. Via thematic analysis (qualitative text analysis of the interviews), we studied what kinds of understandings and expectations the participants formed regarding the thermostat. One striking finding was that the participants developed very different mental models regarding how the thermostats were functioning. The present paper is based on the same field experiment; however, we answer different research questions, and we use different data (mostly quantitative data gathered from the users' interactions with the system).

Hidden Market Design. Seuken et al. [24] argued that for many of the new, complex markets that are emerging to be successful (like the smart grid market), it is a necessity to "hide" some of the market's complexities from the end-users. They proposed the design of a "hidden market user interface (UI)" that makes the interaction with the market more seamless, such that even non-sophisticated users can easily participate in it [25]. To this end, the UI needs to *hide* or reduce some of the interaction complexity for the user. One way to achieve this goal is to design a *learning agent* that operates in the background and mediates between the user and the market. The goal of implementing this agent is to reduce the cognitive costs for the user, while still keeping the important feedback loop between the user and the market that is needed for economic efficiency. In [23], Seuken et al. presented a case study on how to apply hidden market design to the design of a peer-to-peer backup market, demonstrating that it is possible to hide a significant amount of complexity from the end-user, while still keeping the important user–market loop. In [24], Seuken et al. already suggested the smart grid domain as a suitable application area for hidden market design.

Home Heating. One approach aimed at energy-efficient heating is to predict future environmental conditions (e.g., weather) to optimize the heating process. The state-of-the-art method used in the control community is *model predictive control* [17, 18]. In contrast, Shann and Seuken [27] used MDPs to compute a sequentially optimal heating policy given uncertainty about future weather conditions and future electricity prices. An orthogonal approach is to develop algorithms that try to sense and predict the occupancy of the house with the goal of reducing the inside temperature when people are not at home. For example, Scott et al. [22] use motion sensing and machine learning to find patterns in user behavior to heat adaptively. A similar approach is taken by Lu et al. [16]. These approaches are all complementary to the approach taken in this paper and could, in principle, also be included in our thermostat.

Occupancy detection has also been applied in commercial thermostats. For example, the Nest thermostat has a motion sensor that detects people's presence.[2] It learns a heating schedule that conforms to its users' habits. Recently, Nest has started a voluntary demand response program called "Rush Hour Rewards" that remotely controls the air conditioner during peak hours.[3] However, in contrast to our smart thermostat, the Nest thermostat does not learn an individual user's trade-off between comfort and cost.

[2] https://nest.com

[3] https://nest.com/support/article/What-is-Rush-Hour-Rewards

The Underlying Machine Learning Algorithm.

We now briefly describe the learning algorithm introduced by Shann and Seuken [26], as this is the algorithm that we implemented in our smart thermostats. The main components are the *user model*, the *update rule* and the *heating rule*.

User Model. The user's heating preferences are modeled with a *utility function* that quantifies a particular user's trade-off between comfort and cost of heating. Shann and Seuken [26] provide a formula to measure how much utility a user has for a certain indoor temperature at any given price of energy. This utility is composed of a *value* for the indoor temperature minus the *cost* of heating to this temperature. Using this utility function, they derive an individual user's *optimal indoor temperature* at a given price p, which is:

$$T^{opt}(p) = T^* - mp, \tag{1}$$

where T^* is the user's *most preferred temperature* if energy was for free, and $m > 0$ is the user's *sensitivity* to price. Thus, the optimal temperature equation is a weakly decreasing straight line that is defined by the two parameters T^* and m, whose values depend on an individual user's preferences. The linearity of the optimal temperature line follows directly from the assumption of a quadratic loss function regarding the user's preferences (see [26]). This simplifies the model, but is not essential for the system.

Note that the user model assumes that a user behaves in an *economically rational* way upon price changes, i.e., when the price increases then the user is assumed to weakly reduce his temperature. Of course, many different models are plausible to capture a user's trade-off between comfort and cost. For our field experiment, we purposefully chose this relatively simple model, such that the corresponding learning algorithm is robust, and the UI design task (see Section 3.1) was manageable. More sophisticated user models (e.g., [5]) and corresponding learning algorithms could be incorporated into our system, but this is beyond the scope of this paper.

Update Rule. Every time the user changes the setpoint on the thermostat, the algorithm updates its knowledge of the user's preferences. Implicitly, the algorithm assumes that the user solves an optimization problem (how to trade off comfort and cost) when changing the setpoint. To update its knowledge of the user's optimal temperature line (Equation (1)), the learning algorithm uses *Bayesian inference*. The algorithm starts with some prior and treats every new setpoint as *noisy input*, which it uses to compute the *posterior probability* of the optimal temperature. Every time step, it computes the currently optimal temperature $\widehat{T}_{opt}(p_t)$ using the maximum a posteriori estimates of T^* and m. See [26] for details.

Heating Rule. The smart thermostat heats the house in the following way. At every time step t, it sets the setpoint to the estimated optimal temperature for the current price according to the current estimates of the most preferred temperature and the sensitivity:

$$\widehat{T}_{opt}(p) = \widehat{T}^* - \widehat{m}p. \tag{2}$$

3. SYSTEM DESIGN

Figure 1 shows a schematic overview of our system. It consists of the following components: a *Horstmann thermostat*, which is a standard programmable thermostat that can be controlled wirelessly via the z-wave radio protocol; a *Raspberry Pi*, which is a pocket-sized computer on which a z-wave software transceiver is installed that enables communication between the Raspberry Pi and the Horstmann thermostat; a *web application*, which the user can use to remote-control the smart thermostat; and a *web server*.

While the Raspberry Pi controls the setpoint of the Horstmann thermostat, it also receives data regarding the current indoor temperature from the Horstmann thermostat. These two components are

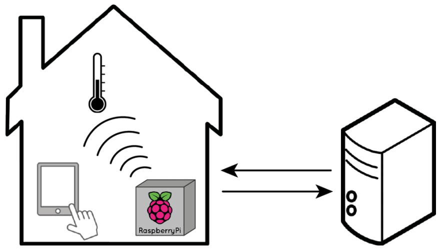


Figure 1: Schematic overview of our smart heating system

installed in a user's home. The Raspberry Pi periodically connects to the web server to pull the latest schedule on how to heat the house for the next several days (based on a particular user's settings). With every pull request, it also sends along the current indoor temperature of the house, which is then stored in a database on the web server. As part of the deployment, users are given a tablet running the web application, which they can use to remote-control the smart thermostat (see Figure 2); alternatively, the users can use any other device with a web browser. In either case, the data for the web application is served by the web server, from which the application also receives the current real-time prices every 30 minutes.

3.1 Design Challenges

The main challenge in designing the UI of the smart thermostat is the inherent tension between the user input and the machine learning output. The learning algorithm assumes the user input (i.e., the setpoint changes) to be *noisy* data. Thus, when the user changes the temperature, the algorithm will update the parameters of the utility function. However, the *optimal temperature according to the model* might be a different value than what the user just provided.

For example, assume the current price is 20 pence/kWh, and the current optimal setpoint according to the user model is 18.5 °C. Assume the user changes the setpoint to 20 °C, and the learning algorithm does a Bayesian update and concludes that the new optimal temperature (based on all previous inputs) is 19 °C. The design challenge is apparent: if the user sets the setpoint to 20 °C, but the system heats to 19 °C instead, then the user will not be satisfied.

We use two different interaction paradigms to reconcile the user input with the machine learning output. The first paradigm is based on *direct manipulation*, exposing the user more directly to how the algorithm is working. The second paradigm lets the user only *indirectly* interact with the learning algorithm. Based on these two interaction modes we designed two UIs, which we call *"learning direct"* and *"learning indirect"*. In addition to these two learning thermostats, we designed a third UI *without machine learning*. In this UI, the user has to manually configure his optimal temperature line. This UI, which we call *"manual"*, served as the control group.

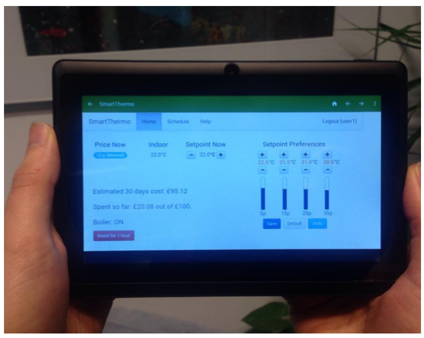


Figure 2: The smart thermostat application running on a tablet

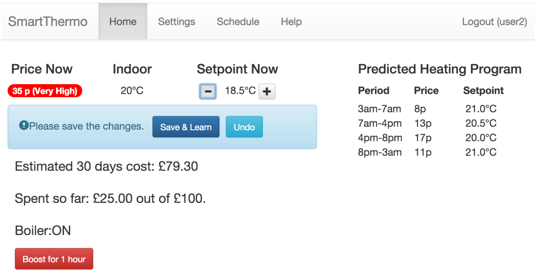


Figure 3: Home page of the “Learning Direct" thermostat

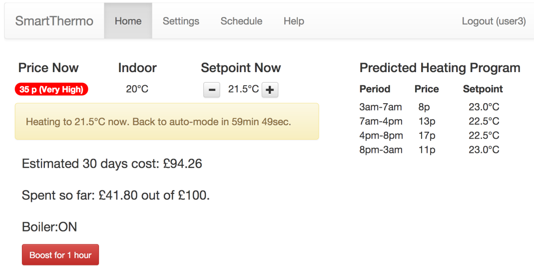


Figure 5: Home page of the “Learning Indirect" thermostat

3.2 The UIs of the Three Thermostats

We first give an overview of the UI elements that are shared by all three versions. For this, consider Figure 3, which shows the home page of the learning direct UI. The page shows the current indoor temperature as well as the setpoint for the current price. The setpoint can be changed by pressing the $+/-$ buttons next to it. The price is color coded (with corresponding labels *normal*, *high*, *very high*) to give the user some intuitive feel for the current price level.

Importantly, we also show the user his *Estimated 30 days cost*, i.e., an estimate how high his heating bill will be, given his current settings. By exploring the financial consequences of different settings, the user can decide how to trade off *comfort* (a warm house) versus *cost* (the monthly heating bill). To compute an estimate of the 30-day costs, we use a simple thermal model of the user's home (see Section 4.1), as well as predictions of the energy prices and the outdoor temperature for the next 30 days. Finally, we also show the user how much of his heating budget he has already spent.

3.2.1 Learning Direct UI

The distinctive feature of the learning direct UI is the fact that *the setpoint that is displayed is always the learned setpoint by the thermostat*. Thus, the semantics of the $+/-$ buttons changes over time. Assume that the current setpoint is 18.5 °C. If the user now presses the warmer button once, the algorithm will take 19 °C as input and do a Bayesian update, resulting in a learned optimal setpoint of 18.7 °C, which is then rounded to 18.5 °C (the granularity is in steps of 0.5 °C). Thus, the user does not see any change in the setpoint. However, if he presses a second time, the algorithm will take 19.5 °C as input and the learned optimal setpoint increases to 18.9 °C, which will then result in a setpoint change to 19 °C. Thus, in this hypothetical example, the user had to press the $+$ button twice to increase the setpoint from 18.5 to 19 °C.

3.2.2 Learning Indirect UI

Figure 5 shows the home page of the learning indirect UI. In this UI, the user is less directly exposed to the machine learning algorithm. The interaction mode for changing the setpoint is as follows. The temperature the user inputs *temporarily overrides* the optimal temperature the algorithm would set. For example, when the user sets the temperature to 20 °C, the thermostat will heat to this exact temperature *for one hour*. In the background, it takes the 20 °C as a new learning input and performs a Bayesian update. After one hour, the thermostat switches to the temperature that will be optimal (according to its new user model) at the then current price.

3.2.3 Manual UI

Figure 4 shows the home page of the manual thermostat. In contrast to the two learning UIs, here the user has to manually specify how the temperature should be set at different prices. He can do this using the *four sliders* on the right side of the UI. The sliders represent the temperature setpoints at 5, 15, 25, and 35 pence/kWh (which covers the whole price range). To maximize the comparability of the manual thermostat with the two learning thermostats, the sliders were constrained to always form a *straight line*, to adhere to the user model underlying the learning algorithm. Thus, if the user changes the setpoint at any slider, the other sliders change their values as well to conform to the linear model.

3.2.4 Settings Page

The settings page (not shown) is an additional screen that is only provided to users of the two learning thermostats. Here, they can review and manage their learned setpoint preferences. The motivation for this screen is to provide an additional level of control for users who are either not satisfied with the price–temperature mapping the thermostat has learned, or who prefer not to interact with the machine learning algorithm. The settings are displayed in the form of four sliders in the same way as on the home page of the manual UI (showing the price–temperature mapping). The user can manually change the temperature on each of the four sliders, and the slider functionality is the same as for the manual UI.

3.2.5 Schedule Page

Our smart thermostat also offers a *schedule page* (see Figure 6) that allows the user to program the heating times of the boiler based on hourly time slots. Here, the user also sees how choosing a particular schedule impacts his estimated 30-days heating cost.

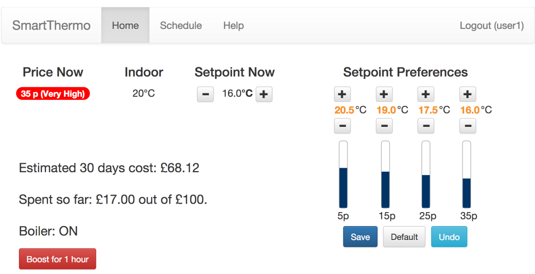


Figure 4: Home page of the “Manual” thermostat

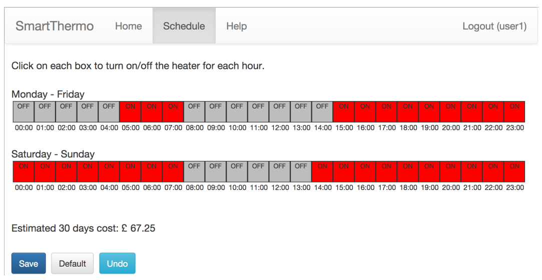


Figure 6: The schedule page

4. EVALUATION

To evaluate our thermostats, we conducted a *field experiment*. We recruited *30 participants* living in England who used the system in their homes for *30 days* from February to March 2015. The participants came from diverse backgrounds, had an average age of 50, and had no prior experience with smart thermostats (see [2] for detailed demographics). We randomly assigned the participants to two *treatment groups* and one *control group*, each with 10 participants. The treatment groups used the learning direct and the learning indirect UI, respectively; the control group used the manual UI.

4.1 Deployment & Incentives

The whole field experiment was divided into a 7-day *data collection phase* and a 30-day *experimental phase*. In a first step, an experimenter went to the users' homes and installed the Horstmann thermostat and the Raspberry Pi. This was followed by the 7-day data collection phase in which we let the users heat their homes normally and recorded their indoor temperature as collected by the Horstmann thermostat. This phase was necessary to personalize the software to each user. In particular, the temperature recordings allowed us to fit the parameter values of the thermal model to each individual home. This served two purposes: first, it created more realism in the study as the predicted heating costs would more closely match the actual costs. Second, it allowed us to create financial incentives tailored to each user (which we will describe shortly).

After the data collection phase, an experimenter visited the users' homes a second time. He instructed the users on how to use the web application using the tablet that was provided (or with any other device running a web browser). Then the actual study with a length of 30 days started. Going forward, every evening, the users were sent a text message to remind them of their current heating budget, their current setpoint, and the current energy price.

Incentives. To create realistic financial incentives, we endowed every participant with a heating budget of £100.[4] We explained to them that they would take part in a *virtual market* for heating in which energy prices change every 30 minutes. We explained that, every day, the heating costs in the virtual market would be subtracted from their virtual heating budget, and at the end of the study, they could keep whatever budget they had left as an experimental reward. Note that it was necessary to simulate the heating costs in a virtual market since nowadays, end-users in the UK do not yet face dynamically changing electricity prices.

Calculating Heating Costs. The calculation of the estimated heating costs was personalized for every user as follows. After the data collection phase, we computed a best fit of the parameters for the thermal model (i.e., leakage rate λ and heater output r_h; see [21]) to the data collected for every user. Furthermore, based on the recorded heating data, we estimated their preferred temperature T^*_{prior}. Finally, we took into account the predictions of the energy prices and the outdoor temperature for the remaining days of the experiment. Given all of this, we then calibrated the heating costs such that heating constantly to $(T^*_{prior} + 1)°C$ *for the whole month* would cost the user £80. Thus, even if the user increased his average setpoint by 1 °C during the experimental phase (and otherwise heated as before), he could still get a £20 reward. Of course, if the user changed his settings, his estimated heating costs changed accordingly. To ultimately calculate the *true* heating costs, we used the same formula, and simply assumed that the heater was on at time t if the recorded temperature was below the current setpoint, and off when the recorded temperature was above the current setpoint. Note that we employed this indirect way to determine when the boiler was on because our system did not have direct access to the boiler.

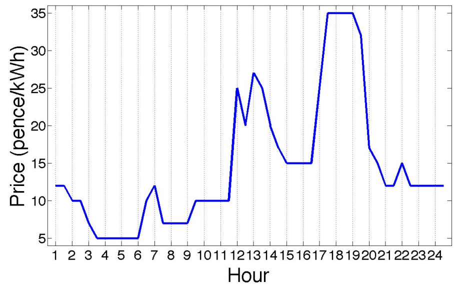


Figure 7: Prices on a sample day

4.2 Prices

To add realism, the prices the users encountered during the study were taken from the UK electricity spot market, dating from January 1 to January 30, 2014.[5] We normalized the prices to range from 5 pence to 35 pence (removing extreme outliers), which resulted in an average price of 12 pence/kWh. The price points are half hourly so that also in the study, prices changed every 30 minutes.[6] While the calculation of the heating cost was personalized to every user, the prices were the same for all users. A *sample* price profile is shown in Figure 7. The prices are low during the night and increase to about the average price level between 8 am and 4 pm. A roughly two-hour long price peak is found between 4 pm and 8 pm, where the price increases around three times compared to the base price. Overall, the price data shows enough variation (intra-day, intra-week, as well as between weekdays and weekends) that we expected the users to face challenging decisions regarding their heating during the study.

4.3 Data Collection

During the study, we gathered both quantitative and qualitative data. We recorded the actual indoor temperature as well as the setpoints every five minutes. In addition to that, we logged all of the users' interactions with the web UI. After the study, we conducted semi-structured interviews with the users. Furthermore, the users filled out a questionnaire with six Likert-scale questions that asked the users to indicate their agreement with a selection of statements on a scale from 1 ("Strongly disagree") to 7 ("Strongly agree").

We analyzed the data in two ways. First, we considered all 30 users. Second, we excluded all users that had fewer than 5 setpoint changes on the home page, leaving us with 21 users (6 in the indirect group, 7 in the direct group, and 8 in the manual group). We call the remaining 21 users the *"active"* users. Whenever it makes sense, we report the results for all users as well as the active users.

5. RESULTS

We now discuss our findings based on the quantitative and qualitative data we collected during the experiment.

5.1 User Experience Analysis

In this section, we first study the user experience of our participants. We ask the following three questions: (1) Did the smart thermostat enable the end-users to handle real-time prices? (2) Did the machine learning algorithm improve the usability of the system? (3) Which of the two learning-based user interfaces worked better?

[4]This corresponds approximately to the amount of money an average UK households spends on *total energy* per month: `https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/487650/table_262.xls`

[5]`https://www.bmreports.com/`

[6]We initialized the study in such a way that the day of the week the prices were taken from corresponded to the day of the week during the study. For example, January 1, 2014 was a Monday; thus, the users saw the prices from this day on a Monday as well.

Overall Satisfaction. Analyzing the interaction logs revealed that all but 3 users interacted with the system at least up to the last week of the study, demonstrating a good level of engagement. The majority of the users seemed relatively happy to delegate control over their heating system to an autonomous system. This is reflected by the average agreement of 5.2 with the sentence: "I trust the thermostat to set the right temperature for me." Users felt in control of their heating and were confident that the system worked correctly. Furthermore, most users seemed satisfied regarding how well they could communicate their heating preferences to the smart thermostat, given the average agreement of 5.4 with the statement "The smart thermostat enables me to express my preferences regarding how to trade off comfort and cost." Overall, the data supports our finding that the smart thermostat achieved its primary goal – to enable users to successfully handle real-time prices. Note that for five out of the six Likert-scale questions, we did not find a statistically significant difference between the three user groups. The only statistically significant result we found was regarding the usability of the system, as we will discuss in the next paragraph.

Usability. We now analyze whether using a machine learning algorithm had a positive effect on usability. Towards this end, we compare the two learning UIs with the manual UI (the control group) regarding the users' average agreement with the statement "The smart thermostat was easy to use." For all 30 users, the averages are 4.9 for learning direct, 5.7 for learning indirect, and 4.0 for manual. A one-way ANOVA finds no significant differences between the three groups ($p = 0.14$). However, for the restricted set of active users, the values are 5.5 for direct, 6.2 for indirect, and 3.3 for manual, and here an ANOVA finds a significant difference between the three groups ($p = 0.01$). Post-hoc comparisons using the Tukey test show that *both learning UIs were rated significantly easier to use than the manual UI*. This supports our original idea of using hidden market design, and in particular to use machine learning, to simplify the interaction with the thermostat.

Comparison of the two Learning UIs. After having seen that the learning feature had a positive effect on the usability, we now compare the two learning UIs and discuss which learning UI was more successful at mediating between the user and the machine learning algorithm (there was no statistically significant difference regarding the users' usability rating of the two UIs). It is important to understand that the two UIs use very different interaction paradigms. Recall that the indirect learning UI temporarily overrides the machine learning output with the user's current setpoint input. This way, the user can easily set the setpoint to any desired temperature - however, after one hour, the setpoint will go back to the learned temperature. In contrast, the direct learning UI always uses and displays the learned setpoint. At the beginning, this may lead to a more "immediate" interaction between the user and the thermostat, because there are not two different temperatures, like with the indirect learning UI. However, after many setpoint inputs have been collected, the learning algorithm starts to converge to a particular setpoint - a natural consequence of the Bayesian updating algorithm. At that time, the $+/-$ buttons on the home page become less reactive. Eventually, if a user provides many inputs (e.g., more than 10), he might need to press the $+/-$ buttons many times until the setpoint changes by 0.5 °C. This might be a source of user frustration. Given this, our hypothesis is that the learning indirect UI was more successful at mediating between the user and the learning algorithm than the learning direct UI. In the following, we present two findings that support this hypothesis.

The first piece of evidence concerns the use of the learning feature. Recall that users of the two learning UIs had two options to change their setpoint preferences: either change the setpoint on the home page, which triggers a Bayesian update, or manually manipulate the sliders on the settings page. Our intention was that people would mostly use the home page to change the setpoint, and only users not satisfied with the learned settings would go to the settings page. To analyze the relative frequency of each setpoint change method, for each user, we measure the ratio $N_{home}/N_{settings}$, where N_{home} is the total number of setpoint changes on the home-page, and $N_{settings}$ is the total number of setpoint changes on the settings page. We remove those users that had zero interactions on the settings page because it would result in a division by zero (two users in each group). Then, the average ratio is 2.6 for learning direct, and 12.9 for learning indirect. A two-sided t-test shows that this difference is statistically significant ($p = 0.02$). Thus, the users of the indirect group used the learning feature much more than the preference changes on the settings page, compared to the users of the direct group.

The second piece of evidence comes from the user interviews. There are at least two users in the direct learning group who complained about the thermostat not changing the setpoint when pressing the $+/-$ buttons:

> P3: "[...] trying to turn the temperature down. Sometimes you'd go down, down, down, down, down, and it doesn't register. And you're going, I pressed down. I pressed down. [...] Wow it needs four presses per half degree or something. [...] So, that was a little bit frustrating [...]"
>
> P10: "It [the thermostat] was more... temperamental. You know you press it sometimes it didn't work"

Summarizing, we state the three main findings of this section. First, users were happy to delegate control over their heating to an autonomous system, which enabled them to successfully handle real-time prices. Second, the learning UIs were rated significantly easier to use than the manual UI, which confirms our hypothesis that hidden market design principles are a valuable tool to design smart grid applications. Third, we presented some evidence that the learning indirect UI was the more successful of the two learning UIs, since it was used as intended and led to a smoother user experience. However, regarding the third point: more research is needed to investigate the optimal design of user interfaces that can effectively mediate between end-users and machine learning algorithms.

5.2 Economic Behavior Analysis

In this section, we discuss our results related to the question how real-time pricing affected the users' economic decision making. In particular, we answer four questions: (1) How did users react to prices changes? (2) Were they willing to reduce their comfort to save money? (3) How much money could they save, and what is the impact of their settings on their comfort? (4) Can we induce a significant amount of demand response during peak hours?

5.2.1 How Do Users React to Price Changes?

Recall that the user model underlying the learning algorithm assumes that people will react to price changes in an "economically rational" way, i.e., when the price increases they will weakly decrease (but not increase) their temperature. Using the real behavior observed in our study, we wanted to verify whether this assumption was ever violated – essentially a sanity check on the model underlying the learning algorithm.

To this end, we analyzed all of the users' setpoint inputs they provided to the system during the study. Each of these data points is a pair (p, T^{set}), where p is the price at which the setpoint T^{set} was saved. We performed the following analysis: given all inputs of

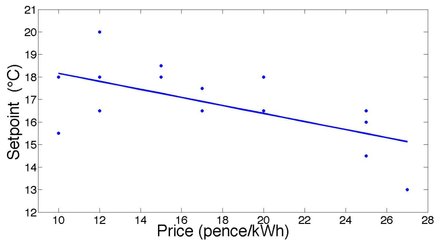


Figure 8: Example setpoint inputs from one particular user, together with best linear fit from linear regression

a user, we ran a linear regression to check for a linear trend in the temperature adjustments. Figure 8 shows an example setpoint cloud from a rational user (each point is a setpoint provided by the user), together with the fitted regression line.

Table 1 summarizes the results of this regression analysis. For 6 out of all 30 users, we find a statistically significant negative slope ($p < 0.05$), which means that their inputs confirm our assumption that people will reduce the temperature if the price increases. For the remaining 24 users, we find slopes that are not statistically significantly different from 0, and thus these users neither confirm nor violate the assumptions of the model (note that the relatively large number of statistically insignificant slopes is largely due to the fact that most users did not provide enough setpoint inputs for the regression to generate statistically significant results). Summarizing, there was no user that violated the rationality assumption of our model, whereas 6 users adjusted the setpoints in a way as predicted by the model. Of course, this does not show that all users acted fully rationally. But it provides us with a certain level of confidence that, at least on average, the basic assumption underlying our model (i.e., that users make trade-offs between comfort and costs) seems reasonable and that our experiment design thus makes sense.

Slope	Direct	Indirect	Manual	Total
Negative ($p < 0.05$)	0	3	3	6
Flat	10	7	7	24
Positive ($p < 0.05$)	0	0	0	0
Total	10	10	10	30

Table 1: "Rationality" Analysis

5.2.2 *User Preferences*

In the previous section, we have analyzed the *stream* of individual setpoint inputs at different prices and at different points in time. In contrast, we now look at the resulting *slope* of the users' optimal temperature lines (whether learned or set manually) at the end of the 30 days, since this slope indicates by how much the users were willing to reduce their temperature when prices were high.

Table 2 shows the users' average slopes; once for all users, and once for all active users, separated by the three groups. None of the differences between the averages are statistically significant. However, the variance of the slopes between the users is noteworthy, varying between -0.31 and 0, which demonstrates the large heterogeneity in the users' preferences.

Slope of T^{opt}	Direct	Indirect	Manual	min / avg / max
All users	-0.11	-0.09	-0.1	-0.31 / -0.1 / 0
Active users	-0.06	-0.11	-0.09	-0.23 / -0.09 / 0

Table 2: The slopes of the optimal temperature lines

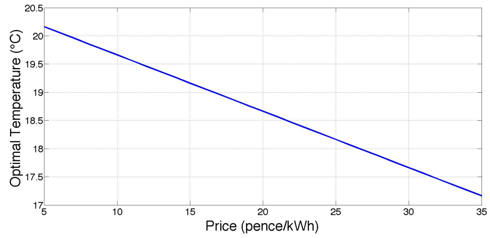


Figure 9: Average optimal temperature line

To visualize what these slopes mean, Figure 9 shows the optimal temperature line for an *average user* with slope $m = -0.1$. The x-axis denotes the price, while the y-axis denotes the optimal temperature. On average, the users' optimal setpoint at 5 pence/kWh was 20.1 °C, and (on average) they were willing to reduce their setpoint to 17.1 °C at 35 pence/kWh, which is a reduction of 3 °C during the price peak. Compare this to the most price-sensitive user, who had a slope of $m = -0.31$. Thus, he was willing to reduce his temperature by 9.3 °C during the price peak. Note, however, that this particular user had a very high "most preferred temperature" of $T^* = 26.5$ °C. Thus, when prices were low, he was heating to a very high setpoint, but when prices were high, then this user was willing to radically reduce his temperature – in theory to 17.2 °C.

However, as we will discuss in the next section, even though some users had a very large *willingness* to reduce their temperature when prices were high, the actual temperature drop during price peaks was much smaller, due to the thermal inertia of most homes.

5.2.3 *Comfort-Cost Trade-off*

We have seen that, on average, the users' thermostat settings suggest that they were willing to sacrifice some of their thermal comfort to save some money. The questions that follow from this observation are: how much money did they actually save, and how did their settings actually influence the temperature in their homes?

Cost Analysis. Table 3 summarizes the total cost data. On average, the users' heating costs (over 30 days) were £47. Thus, at the end of the study, they had an average of £53 left from the £100 heating budget. While the learning indirect group had lower costs than the other two groups, this difference is not statistically significant ($p = 0.06$). The most likely explanation for this difference is a difference in the heating schedules. The learning indirect users heated least (6.4 hours per day on average, weighted over work days and weekends), while the learning direct and the manual users heated more (9.3 and 9.2 hours per day, respectively). Note, however, that this difference is also not statistically significant ($p = 0.17$), but still big enough to have an observable impact on the costs.

Total cost	Direct	Indirect	Manual	min / avg / max
All users	£55	£32	£55	£14 / £47 / £100

Table 3: Participants' total heating cost over 30 days

Comfort Analysis. Note that, even though the users *allowed* the smart thermostat to decrease the setpoint by 3 °C on average during price peaks, the *actual* temperature drop was much smaller due to the thermal inertia of a home and the limited duration of the peak. In the context of our study, we define a peak to be an event during which the price stays above twice the average price of 12 pence/kWh for at least 2 hours. A duration of 2 hours is interesting because only if the peak is long enough, then the users are expected to experience the impact of the temperature settings on their comfort. Using this definition, we identify four price peaks in our study.

Clearly, the greatest impact on users' comfort happens during these four price peaks. However, when we analyze the temperature data (focusing on the *active users*), we see that even during these peak events, the comfort loss was within acceptable bounds. Only in 2 (10%) of the homes the temperature fell by 1 °C during the peaks, while in 13 homes (62%), the temperature did not change. The temperature in the remaining 6 homes (28%) increased by 1 °C during the peaks.[7] This indicates that most users did not suffer from big temperature drops even during price peaks.[8]

5.2.4 Demand Response Analysis

When designing a smart thermostat to enable demand-side management, it is important to note that the overall goal of demand-side management is to reduce the demand during *peak hours*, i.e., during price peaks that last a significant amount of time [3]. This can be looked at from two perspectives: the *users' perspective* and the *network operator's perspective*. We have already covered the user's perspective in the previous sections, i.e., how much they are willing to decrease the setpoint during price peaks, and, importantly, how much actual comfort loss they will suffer for doing so.

For the network operators, the goal is to reduce the demand for energy during peak hours. A common metric used for evaluating demand response programs is the normalized actual *demand reduction* which measures the percentage reduction in energy consumption during price peaks [3], and is defined as

$$DR = \frac{C_{offpeak} - C_{peak}}{C_{offpeak}},$$

where C_{peak} is the consumption that was actually measured during the peak, and $C_{offpeak}$ is the hypothetical (baseline) consumption *that would have been measured* had there been off-peak prices instead.[9] We use the same definition of peak as in the previous section. As the baseline (i.e., $C_{offpeak}$), we take the consumption that would have occurred if the price had stayed at 12 pence/kWh instead. Since only C_{peak} is observed, $C_{offpeak}$ must be estimated. However, due to the relatively short duration of the study and the high variation in each user's settings and occupancy patterns, it was not possible to reliably estimate the counterfactual "off-peak demand" from the experimental data. For this reason, we used our simulation model for this estimation. To this end, for every user, we use the thermal model of the user's house, the user's setpoint preferences and his schedule at the time of the peak, to estimate what this users's consumption would have been at the same time when the price peak occurred, but assuming a constant price of 12 pence/kWh instead.

Using this approach, we estimate the average demand reduction to be DR = 38% (Table 4 provides additional results). Interestingly, when considering the set of active users, we obtain 50% of demand response via the indirect UI, and this is almost twice as large as the demand response achieved by the direct UI (27%). However, this difference is not statistically significant (p=0.097).

Comparison to other Trials. Compared to other demand response trials from the literature, the amount of demand response we found (DR=38%) is relatively large. A meta-study by Stromback et al. [28] found that, using automation technology, an average reduction of 21% can be achieved. The study evaluated 85 field pilots conducted in the US, Canada, Europe, and Japan. Apart from real-time pricing, these pilots also tested times-of-use tariffs and critical peak pricing. The study found that critical peak pricing generally leads to the highest amount of demand response (31% on average). A notable example is Gulf Power's residential service variable pricing pilot in Florida [7]. Their customers could program their thermostats to automatically react to the current electricity price, similarly to our smart thermostat. The average demand response during critical price periods (where the price was approximately 5 times the average price) was estimated to be 41%. This matches our finding that high amounts of demand response can indeed be achieved in the residential sector with automation technology.

Demand response	Direct	Indirect	Manual	min / avg / max
All users	34%	47%	36%	0% / 38% / 100%
Active users	27%	50%	35%	0% / 36% / 100%

Table 4: Demand response analysis

6. LIMITATIONS

Our work has a number of limitations. First, we use the indoor temperature as a proxy for a user's comfort, although thermal comfort is a complex phenomenon that depends on many variables [4]. We decided to use the indoor temperature as a proxy for comfort because it is a very important factor influencing comfort and because it is simple and robust to measure. A second limitation concerns the thermal heating model that we employed. While this model has been validated by prior research [21], it is a relatively simple model, and there do exist more complex models, capturing the thermal properties of buildings and the physical process of heating more accurately. However, the purpose of using a thermal model in our study was not to provide the most accurate 30-day cost prediction possible, but to create enough realism such that the users could immerse themselves into the scenario of heating with real-time prices.

7. CONCLUSION

The goal of this research project was to design a smart thermostat that enables users to handle home heating in a real-time pricing regime. We followed the hidden market UI design approach and built an autonomous heating system that automates the heating by responding to price signals on a user's behalf and learns a user's comfort-cost trade-off over time. We tested two designs of the learning thermostat against a non-learning, manual, baseline in a field experiment in the UK with 30 users over a period of 30 days.

Our results show that the smart thermostat enabled users to deal with real-time prices, leading to a large amount of demand response while keeping users' comfort within acceptable bounds even during price peaks. Furthermore, the learning UIs were rated significantly easier to use than the manual one, which confirms the value of hiding some of the interaction complexity from the user.

Overall, we conclude that it is possible to induce a large amount of demand response even with a small amount of interaction. This suggests that smart (learning) thermostats could provide a viable alternative for users that prefer less complex user interactions.

Acknowledgments

We thank Alex Rogers for insightful discussions and the anonymous reviewers for helpful comments. This work was partially supported by the SNSF (Swiss National Science Foundation) project #147211, by the EPSRC ORCHID project (EP/I011587/1), and through a PhD scholarship from the Turkish Government. Study approved by University of Southampton FPSE Ethics Committee (ref: 13417).

[7] The temperature can increase during a price peak for multiple reasons. For example, for users with zero slope, prices have no effect. But even for price-sensitive users, their heating may coincidentally be scheduled such that it happens to start heating in the middle of a price peak, and then the boiler may be on despite high prices.

[8] Note that the precision of the thermostat is 1 °C and therefore, we cannot present more exact data.

[9] An alternative measure that is used in the context of real-time pricing is the *price-elasticity of demand* [29]. We do not use it because we are interested in the actual reduction during price peaks.

REFERENCES

[1] A. Alan, E. Costanza, J. Fischer, S. D. Ramchurn, T. Rodden, and N. R. Jennings. A field study of human-agent interaction for electricity tariff switching. In *Proceedings of the International Conference on Autonomous Agents and Multi-agent Systems*, pages 965–972, Richland, SC, 2014.

[2] A. T. Alan, M. Shann, E. Costanza, S. D. Ramchurn, and S. Seuken. It is too hot: An in-situ study of three designs for heating. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 5262–5273, San Jose, CA, 2016.

[3] M. Albadi and E. El-Saadany. Demand response in electricity markets: An overview. In *Power Engineering Society General Meeting, 2007. IEEE*, pages 1–5, June 2007.

[4] ASHRAE. Standard 55 – Thermal Environmental Conditions for Human Occupancy, 2013.

[5] F. Auffenberg, S. Stein, and A. Rogers. A personalised thermal comfort model using a bayesian network. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2547–2553, 2015.

[6] G. Barbose, C. Goldman, and B. Neenan. *A Survey of Utility Experience with Real Time Pricing*. Lawrence Berkeley National Laboratory, December 2004.

[7] S. Borenstein, M. Jaske, and A. Rosenfeld. Dynamic pricing, advanced metering, and demand response in electricity markets. *Center for the Study of Energy Markets*, 2002.

[8] J. Bourgeois, J. van der Linden, G. Kortuem, B. A. Price, and C. Rimmer. Conversations with my washing machine: An in-the-wild study of demand shifting with self-generated energy. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*, pages 459–470, Seattle, WA, 2014.

[9] Committee on Climate Change. Fourth Carbon Budget Review. Technical Report, December 2013.

[10] E. Costanza, J. E. Fischer, J. A. Colley, T. Rodden, S. D. Ramchurn, and N. R. Jennings. Doing the laundry with agents: A field trial of a future smart energy system in the home. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 813–822, Toronto, Canada, 2014.

[11] P. Cramton and A. Ockenfels. Economics and design of capacity markets for the power sector. *Zeitschrift fuer Energiewirtschaft*, 36(2):113–134, 2012.

[12] A. Faruqui and J. Palmer. Dynamic Pricing of Electricity and its Discontents. *Regulation*, 34(2), 2011.

[13] D. Hammerstrom, R. Ambrosio, J. Brous, T. Carlon, D. Chassin, J. DeSteese, R. Guttromson, G. Horst, O. Järvegren, R. Kajfasz, et al. Pacific Northwest Gridwise Testbed Demonstration Projects. *Part I. Olympic Peninsula Project*, 2007.

[14] S. Kärkkäinen, J. Ikäheimo, C. Kofod, D. Giraud, H. Nordvik, and O. Grande. Energy Efficiency and Load Curve Impacts of Commercial Development in Competitive Markets-Results from the EFFLOCOM Pilots: VTT, Finnland. 2004.

[15] C. King. PowerCentsDC Program Final Report. Smart Meter Pilot Program, Inc., eMeter Corp. Washington, DC, 2010.

[16] J. Lu, T. Sookoor, V. Srinivasan, G. Gao, B. Holben, J. Stankovic, E. Field, and K. Whitehouse. The smart thermostat: Using occupancy sensors to save energy in homes. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, pages 211–224, Zurich, Switzerland, 2010.

[17] Z. Y. L. McLaughlin, L. Jia, M. C. Murphy-Hoye, A. Pratt, and L. Tong. Modeling and Stochastic Control for Home Energy Management. In *Proceedings of the Power and Energy Society (PES) General Meeting*, 2012.

[18] F. Oldewurtel, A. Parisio, C. N. Jones, M. Morari, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, and K. Wirth. Energy Efficient Building Climate Control Using Stochastic Model Predictive Control and Weather Predictions. In *Proceedings of the American Control Conference*, 2010.

[19] J. Palmer and I. Cooper. Great Britain's Housing Energy Fact File. Departmant of Energy and Climate Change (DECC), September 2013.

[20] S. D. Ramchurn, P. Vytelingum, A. Rogers, and N. R. Jennings. Putting the 'smarts' into the smart grid: A grand challenge for artificial intelligence. *Communications of the ACM*, 55(4):86–97, April 2012.

[21] A. Rogers, R. Wilcock, S. Ghosh, and N. R. Jennings. A scalable low-cost solution to provide personalized home heating advice to households. In *Proceedings of the 4th ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings (BuildSys)*, pages 211–212, Toronto, Canada, 2012.

[22] J. Scott, A. Bernheim Brush, J. Krumm, B. Meyers, M. Hazas, S. Hodges, and N. Villar. Preheat: Controlling home heating using occupancy prediction. In *Proceedings of the 13th International Conference on Ubiquitous Computing (UbiComp)*, pages 281–290, Beijing, China, 2011.

[23] S. Seuken, D. Charles, M. Chickering, and S. Puri. Market design analysis for a P2P backup system. In *Proceedings of the 11th ACM Conference on Electronic Commerce (EC)*, pages 97–108, Cambridge, MA, 2010.

[24] S. Seuken, K. Jain, and D. C. Parkes. Hidden market design. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pages 1498–1503, Atlanta, GA, 2010.

[25] S. Seuken, K. Jain, D. S. Tan, and M. Czerwinski. Hidden markets: UI design for a P2P backup application. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 315–324, Atlanta, GA, 2010.

[26] M. Shann and S. Seuken. An active learning approach to home heating in the smart grid. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2892–2899, Beijing, China, 2013.

[27] M. Shann and S. Seuken. Adaptive Home Heating under Weather and Price Uncertainty using GPs and MDPs. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, Paris, France, 2014.

[28] J. Stromback, C. Dromacque, and M. H. Yassin. The potential of smart meter enabled programs to increase energy and systems efficiency: a mass pilot comparison short name: Empower demand. *Vaasa ETT*, 2011.

[29] L. Summit Blue Consulting. Evaluation of the 2005 Energy-Smart Pricing Plan-Final Report. 2006.

[30] U. S. Department Of Energy. Grid 2030: A National Vision For Electricity – Second 100 Years. 2003.

[31] U.S. Energy Information Administration. Energy Consumption in Homes by End Users. http://www.eia.gov/consumption/residential, March 2013.

[32] R. Yang and M. W. Newman. Learning from a learning thermostat: Lessons for intelligent systems for the home. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*, pages 93–102, Zurich, Switzerland, 2013.

Curriculum Vitae

Personal Information	
Name	Michael Shann
Date of Birth	July 20, 1983
Place of Birth	Zurich, ZH, Switzerland
Nationality	Swiss

Professional Experience	
02/2019 -- present	*Software Engineer*, Amstein + Walthert AG
04/2016 -- 01/2019	*Software Engineer*, dormakaba AG

Education	
2011 -- 10/2019	*Ph.D. studies*
	Computation and Economics Research Group
	Department of Informatics
	University of Zurich
2008 -- 2011	*MSc in Computer Science*
	ETH Zurich
2005 -- 2008	*BSc in Computer Science*
	University of Zurich
2004 -- 2005	*Classical Philology,* Linguistic access passed
	University of Zurich
2003 -- 2004	*Physics,* 1. Vordiplom passed
	ETH Zurich
2002	*Matura*
	Kantonsschule Hohe Promenade, Zurich, ZH